Theses and Dissertations | 1. Thesis and Dissertation Collection, all items

2017-03

Mapping ad hoc communications network of a large number fixed-wing UAV swarm

Pospischil, Alexis

Monterey, California: Naval Postgraduate School

http://hdl.handle.net/10945/53035

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

MAPPING AD HOC COMMUNICATIONS NETWORK OF A LARGE NUMBER FIXED-WING UAV SWARM

by

Alexis Pospischil

March 2017

| Thesis Co-Advisors: | Duane Davis |
| | Justin Rohrer |

Approved for public release. Distribution is unlimited.

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704–0188*

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE March 2017	3. REPORT TYPE AND DATES COVERED Master's Thesis 06-01-2015 to 03-31-2017	

4. TITLE AND SUBTITLE MAPPING AD HOC COMMUNICATIONS NETWORK OF A LARGE NUMBER FIXED-WING UAV SWARM	5. FUNDING NUMBERS
6. AUTHOR(S) Alexis Pospischil	

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943	8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

In 2015, a group of Naval Postgraduate School (NPS) professors and students set a record when they flew 50 fixed-wing unmanned aerial vehicles (UAVs) simultaneously as a self-organizing swarm. These vehicles were able to execute behaviors based on message notification from a single ground station, and then decide within their swarm group how to order themselves. They were able to accomplish this by communicating over their 802.11n wifi connections. Understanding the strengths and weaknesses of this network will be essential to scaling the swarm to larger sizes or even creating partitioned sub-swarms. The work covered in this thesis is to build a model of the NPS swarm's communication network in ns-3 simulation software and use popular network metrics to illustrate the performance of the network as swarm size increases. It also applied four routing protocols to the swarm and compares their performance to the broadcast protocol. The swarm's communication network was not very tolerant of overhead. This thesis concludes that any routing protocol applied to the (NPS) swarm in the future should consider protocols that reduce or strictly manage overhead generated by either routing tables or multiple message copies. Goodput and packet delivery ratio were the quantitative metrics used. While they illustrate reliability, they do not give a good picture of latency. It would be useful to add latency as a quantitative metric to future work because some swarm messages are more time-sensitive than others. It may be that more than one routing protocol or a protocol with variable settings would be best for this swarm and its various message priorities.

14. SUBJECT TERMS UAV communications network, swarm communications network, UAV swarm, fixed-wing UAV swarm, UAV, swarm	15. NUMBER OF PAGES 73
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU

Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

**MAPPING AD HOC COMMUNICATIONS NETWORK OF A LARGE NUMBER
FIXED-WING UAV SWARM**

Alexis Pospischil
Lieutenant, United States Navy
B.S., University of South Carolina, 2003

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

**NAVAL POSTGRADUATE SCHOOL
March 2017**

Approved by: Duane Davis
                    Thesis Co-Advisor

                    Justin Rohrer
                    Thesis Co-Advisor

                    Peter J. Denning
                    Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In 2015, a group of Naval Postgraduate School (NPS) professors and students set a record when they flew 50 fixed-wing unmanned aerial vehicles (UAVs) simultaneously as a self-organizing swarm. These vehicles were able to execute behaviors based on message notification from a single ground station, and then decide within their swarm group how to order themselves. They were able to accomplish this by communicating over their 802.11n wifi connections. Understanding the strengths and weaknesses of this network will be essential to scaling the swarm to larger sizes or even creating partitioned sub-swarms. The work covered in this thesis is to build a model of the NPS swarm's communication network in ns-3 simulation software and use popular network metrics to illustrate the performance of the network as swarm size increases. It also applied four routing protocols to the swarm and compares their performance to the broadcast protocol. The swarm's communication network was not very tolerant of overhead. This thesis concludes that any routing protocol applied to the (NPS) swarm in the future should consider protocols that reduce or strictly manage overhead generated by either routing tables or multiple message copies. Goodput and packet delivery ratio were the quantitative metrics used. While they illustrate reliability, they do not give a good picture of latency. It would be useful to add latency as a quantitative metric to future work because some swarm messages are more time-sensitive than others. It may be that more than one routing protocol or a protocol with variable settings would be best for this swarm and its various message priorities.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

AODV   Ad hoc On-Demand Distance Vector

ARP   address resolution protocol

ASCII   American Standard Code for Information Interchange

DSDV   Destination-Sequenced Distance-Vector Routing

DSR   Dynamic Source Routing Protocol

DTN   Delay (or Disruption) Tolerant Network

FANET   flying ad hoc network

GAPR   Geolocation Assisted Predictive Routing

GPS   Global Positioning System

Hz   hertz

ICMP   internet control message protocol

IP   Internet Protocol

kg   kilogram

km   kilometers

m   meters

m/s   meters per second

MANET   Mobile ad hoc network

NPS   Naval Postgraduate School

OLSR   Optimized Link State Routing

PDR   Packet Delivery Ratio

Prophet   Probability Routing Protocol using History of Encounters and Transitivity

RC   radio-controlled

s   seconds

SMAVNET   swarming micro air vehicle network

t   time

TCP   Transmission Control Protocol

UAV   unmanned aerial vehicle

UDP   User Datagram Protocol

ZRP   Zone Routing Protocol

Acknowledgments

I wish to extend my sincere thanks to my advisers, Dr. Justin Rohrer and Dr. Duane Davis. Through their support and guidance, I was able to pursue a thesis topic that was both interesting and at the forefront of unmanned aerial vehicle (UAV) technology. It was quite challenging but they offered their time and expertise whenever I felt lost.

I also want to thank my colleagues; I will remember them long after I graduate. I especially want to mention Dan Lukaszewski, Boulat Chainourov, and Anthony Ambriz for their patience, their time, and their friendship.

Finally, I want to thank my family for their love and support. They have been there from the moment I joined the military ready to send encouragement, care packages, or act as a sounding board for my many reports and projects. It feels as if we made it through school together. Now, on to our next adventure.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

The creators of science fiction not only capture the readers' imagination, but also inspire inventors, engineers, and scientists to make the fantastical into reality. Flip phones were modeled after the "Star Trek" communicators, ear bud headphones first appeared as a concept in *Fahrenheit 451*, debit/credit cards providing instantaneous access to your money were first described in *Looking Backward*, and robot swarms capable of self organization have been depicted in the novel *Prey*, by Michael Crichton and television shows like "Agents of S.H.I.E.L.D". Inspiration can come from the imaginative minds of people or from the world around us. Swarms have demonstrated a high degree of success. Bees, ants, termites, and naked mole rats maintain large groups that distribute tasks among individuals in order to achieve great things for the success of the colony. Fish, birds, and bats move and swirl in great numbers without colliding. Man has sought to reproduce these working models.

One key aspect that the natural models share is a form of reliable communication between individuals. Information at the small scale builds to achieve goals at the large scale. For a swarm of aerial vehicles, that communication is most likely over some form of 802.11 network. This thesis models the network currently used by a swarm of fixed-wing aerial vehicles developed at the Naval Postgraduate School (NPS) and compares its performance to Delay (or Disruption) Tolerant Networks (DTNs) in order to determine the best routing protocol for the current swarm configuration as well as future larger scales.

1.1 The Swarm

In 2015, a group of NPS professors and students set the record for largest fixed-wing unmanned aerial vehicle (UAV) swarm flown at one time [1]. The swarm had 50 vehicles flying simultaneously and successfully demonstrated distributed decision-making with all processing occurring on swarm vehicles rather than a centralized control station.

The NPS swarm uses custom messages coded at the application level for vehicle-to-vehicle and vehicle-to-ground communications. User Datagram Protocol (UDP) broadcast is used to transmit these messages over an 802.11n ad hoc wireless network. This information

exchange protocol was chosen to prioritize low latency over reliability, but research had yet to be done to determine if it is ideal for large-swarm communication or if other options are better. Of particular concern is the characterization of communications requirements as the swarm continues to scale up to larger numbers.

The NPS swarm relies on several types of messages in order to function, a number of which are transmitted at regular intervals. For safety reasons, a heartbeat message is sent from the ground station to the vehicles at an interval of 1 hertz (Hz). If a vehicle does not receive that message for a period of 30 seconds, it aborts its current mission and loiters at a pre-designated point to attempt to re-establish its link with the ground station. If the vehicle does not receive an update within two minutes, it executes its autonomous landing procedure [1].

Individual vehicles send flight status messages to the ground station at a rate of 2 Hz. This message updates the ground controller on the health of that particular vehicle (e.g., battery life and autopilot mode). Vehicles also send pose messages to update the ground station and other swarm vehicles as to their current state at a rate of 10 Hz [1].
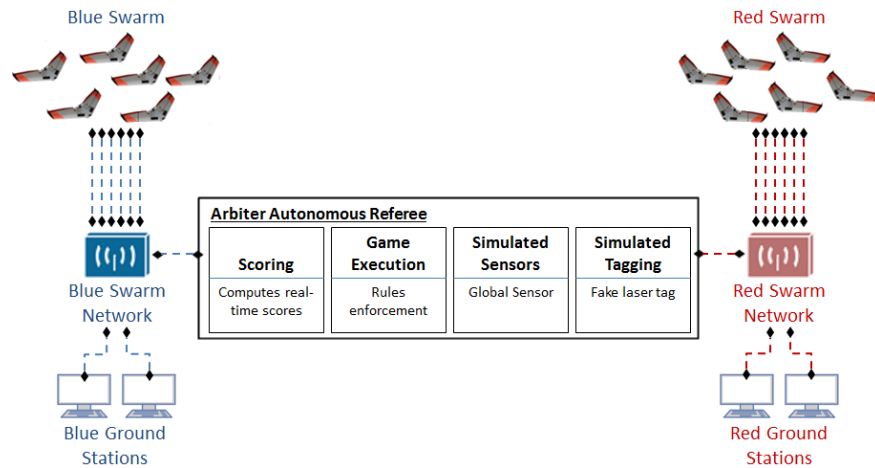
Figure 1.1: Arbiter configuration for bridging red and blue networks and refereeing the ACS Challenge. Source: [2]

The last synchronous message is the red pose message, transmitted by a special purpose ground station, the arbiter, as a means of providing "virtual sensor" information during competitive multi-swarm events [2]. These messages are transmitted by the arbiter upon receipt of pose messages from one swarm to the adversarial swarm operating on a different network as depicted in Figure 1.1 from [2]. Since pose and red pose messages are sent at

a regular-interval frequency (10 Hz), the rate of these messages will scale linearly with the number of UAVs after accounting for packet loss rate of the underlying 802.11n network.

In addition to these synchronous messages, a number of asynchronous messages are provided. These messages address a number of special purpose requirements such as assignment of vehicles to sub-swarms for tasking, initiation and termination of swarm behaviors, direction or parametrization of individual vehicle actions, and the exchange of behavior-specific information between vehicles [1]. These asynchronous messages can be broadcast to the entire swarm or directed to a specific vehicle or ground station (although still transmitted as a UDP broadcast message). Messages directed to a single vehicle or ground station can utilize a "reliable" mode that provides for retransmission of the message until an acknowledgment is received from the intended recipient [2]. These messages are used to provide direction to a particular vehicle (e.g., initiate or terminate a behavior) or exchange critical information between vehicles (e.g., subtask assignments within a swarm behavior). These acknowledgement messages are generated and sent at the application layer and should not be confused with Transmission Control Protocol (TCP) acknowledgements on the transport layer [1].

Given their frequency and contents, synchronous messages comprise the bulk of the NPS swarm communications requirement. This work will therefore focus on analysis of synchronous message traffic within the NPS swarm.

The messages are vital to the performance of the swarm. Given adequate communication performances, individual UAVs no longer need a pilot to control most flight details and distributed swarm computation allows a single operator to efficiently and safely direct the activity of large numbers of vehicles. Furthermore, autopilot software has advanced to the point where a user can select waypoints and the vehicle will maneuver itself to that point in the sky. A single ground station can then be used to orchestrate the behaviors of 50 vehicles using simple and infrequent messages as long as messages are received by the intended destination in a timely manner. Currently, all NPS swarm messages are sent via an omnidirectional broadcast from the source using the UDP Internet Protocol (IP) and the 802.11n wireless networking standard, which provides for unreliable delivery to all communications nodes within the network. Upon receipt, all entities utilize the application layer protocol to determine whether to accept and process the message or drop it. Received

3
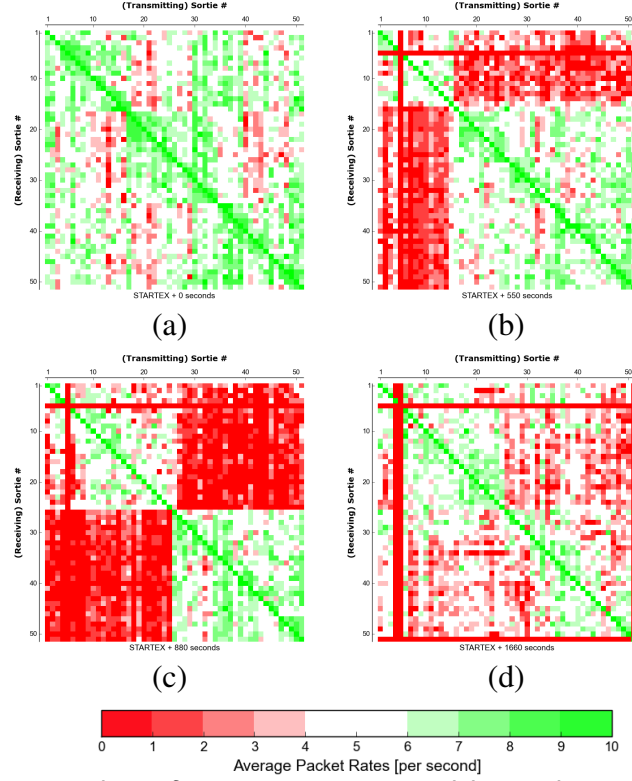
Figure 1.2: Network performance as measured by packet rate between aircraft, i.e., packet rate observed at UAV i (row) received from UAV j (column), over the course of the 50-UAV flight test. Packet rates shown: (a) at start of the experiment ($t = 0$), where all 50 aircraft are still on-deck; (b) after the first 15 UAVs have been launched ($t = 550$ seconds); (c) after the first 25 UAVs are airborne into Stack 1 ($t = 880$ seconds); and (d) once all 50 UAVs are aloft ($t = 1660$ seconds). Source: [1]

messages are accepted if they are directed to the receiving entity (point-to-point) or the entire swarm (swarm-wide broadcast). No multi-packet flow between source and destination is ever established. No message is ever acknowledged at the transport layer. As stated earlier, messages utilizing "reliable" mode compel the vehicle to generate and send an acknowledge message, which is handled at the application layer and not the transport layer.

Real-world vehicle-to-vehicle packet delivery rates for the 50-UAV swarm event were described in [1]. In this paper, Figure 1.2 provides typical packet delivery rates between each of the 50 aircraft on a scale ranging from 0 to 10 packets per second. When all aircraft were on the ground at time (t) = 0 seconds (s), packet delivery rates covered the high end of the scale (6 – 10 packets per second). At t = 550s, 15 UAVs had been launched. Packet

delivery rates between UAVs aloft and UAVs on the ground have packet delivery rates that are mostly 0 – 3 packets per second while UAVs communicating between other UAVs with the same flight status (aloft vs. grounded) maintain the same high rate of packet delivery as recorded at t = 0s. The same was true at t = 880s where 25 of the 50 aircraft were aloft. At t = 1660, all aircraft were aloft. Packet delivery rates during this mission phase were relatively high (i.e. in the 6-8 range) for UAVs in the same subswarm but markedly lower (e.g., in the 2-5 range) for UAVs in different subswarms. This provides anecdotal depiction of the effect of geographic separation between subswarms performing different tasks.

Rohrer and Jabbar [3] examine the shortfalls of using a TCP/UDP/IP in an aeronautical environment. Nodes are highly mobile, which results in end-to-end paths being short-lived. The TCP routing protocol was designed for long-lasting connections along an established path. Attributes of TCP, like the three-way handshake, slow start algorithm, and congestion control algorithm, do not allow this routing protocol the flexibility to deliver packets quickly in an aeronautical node environment. UDP is not encumbered by the same restraints as TCP but offers no acknowledgment that packets were delivered to the destination.

1.2   Problem Statement

This research constructed a model of the ad hoc communication network used by a large number of fixed-wing UAVs flown by NPS faculty and students in order to determine the best parameters by which to measure successful packet delivery, and compare its performance to other network protocols that might provide better performance.

1.3   Thesis Organization

Chapter 2 will review the issues currently being tackled in UAV research and development as well as a review of routing protocols used in the simulator or that should be considered in future work. Chapter 3 will cover the methodology used in constructing the model of the NPS swarm and how it was tested. Chapter 4 is a synopsis of the test results supported by graphs that illustrate the relationships between both protocols and swarm sizes. Finally, Chapter 5 contains the conclusions drawn from the results as well as suggestions for future work.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Background and Related Works

This chapter introduces the necessary concepts of UAV swarm research; how it has progressed, common issues, and research towards finding solutions to these issues. Very often, dynamic networks have properties unique to their environment and/or purpose. Solutions for one group may not work for another without changes to the routing protocol. As a result, there are many variations to the basic founding ideas to consider.

2.1 Swarm Networking Issues

Aerial swarms have the capacity to move quickly, not only in relation to the ground and base station, but also in relation to other vehicles. The swarm is composed of the NPS-designed and built Zephyr II UAV, which is a 2.5 kilogram (kg) fixed-wing vehicle based on a commercially available platform from Ritewing. It has a wingspan of 1.45 meters (m) and a cruising speed of 18 meters per second (m/s) [1]. The NPS swarm uses no routing protocol. Rather, it broadcasts all messages over the 802.11n wireless network and relies on the vehicles to determine from the application-level message header whether to drop a message not intended for that vehicle, or accept the message and respond accordingly.

R. Stefano et al. [4] implemented a fixed wing swarm using a somewhat different communications approach. This swarm consisted of eBee fixed-wing UAVs made by SenseFly. These UAVs have a wingspan of 0.96 m and a cruising speed of about 16 m/s. It was noted that this high mobility can create a highly dynamic topology wherein links that previously existed between individual vehicles are frequently broken. Stefano et al. addressed this issue with a routing protocol that uses knowledge of Global Positioning System (GPS) data to predict the best routing path over the 802.11n wireless network.

Another issue that arises for mobile aerial vehicles is the ability for them to travel away from the base station and/or other vehicles. This allows for a larger operating area as vehicles disperse throughout an unbounded three-dimensional space, which may benefit the mission. This can result in high density areas which contrast with areas of low density where only one or two vehicles may be within range of one another. A vehicle or small group of vehicles

may even lose connectivity with the larger body of the swarm for a period of time. Swarms must be resilient to these topology changes. Base stations typically have sufficient power supplies to transmit longer range communications to vehicles, but vehicles are limited by the size of the battery they are able to carry. Power must be split between communications, flight, and sensors. In any case, until aerial swarms are able to scale well, they will have lower node density than sensor networks to which they are sometimes compared [5].

Bandwidth is yet another constraint in aerial vehicle swarms. When all the nodes are trying to send messages, and the number of nodes is significant, bandwidth can be overwhelmed. Message conveyance is not the only stress on bandwidth, though. Overhead from routing protocols can be the most significant taxation on bandwidth. Section 3.5 provides more detail about the kinds of overhead produced by multi-hop routing protocols.

2.2 Current Swarm Networking Tactics and Models

The NPS swarm does not implement multi-hop routing protocols. The base station and every vehicle send messages over the 802.11n wireless network using UDP, which does not require an end-to-end connection, does not require acknowledgement of packet delivery, and does not control a congestion window [6]. This saves the network vital bandwidth by avoiding overhead in acknowledgements and/or network convergence, saves time by not waiting for end-to-end connections that may last for short windows, and does not slow the rate of messages if packets to not arrive. In fact, the messages are all implemented at the application layer and only use the link layer for conveyance. The vehicle applications are responsible for accepting messages and sending replies when appropriate.

This makes for a low latency networking scheme when all entities are within broadcast range. Where the swarm runs into problems is scaling and future subswarm missions. Fifty vehicles is the current highest concentration of vehicle achieved using broadcast overUDP with application layer messaging. It is not currently known how many vehicles will bring the network to saturation nor how far the vehicles may travel before losing the ability to communicate with the ground station. Vehicles have been flown as far as 2 kilometers (km) away and 850 m of elevation away from the ground station (at the limits of the testing grounds) without losing connectivity. As mission behaviors become more advanced, it may be necessary to test these limits. At this point, a routing protocol that does not overly stress

the bandwidth but allows intermediary vehicles to relay messages through the swarm may prove beneficial. Examples of other swarms and their techniques include:

- 2004: Three quadcopter UAVs using GPS navigation and a Bluetooth connection with a ground station that is able to adjust positioning data, and send navigation commands as waypoints. Commands are given to each vehicle through an established end-to-end Bluetooth connection. Vehicles are not able to exchange information [7].

- 2013: Three fixed-wing UAVs in a leader-slave-slave relationship. Each vehicle is given updated information about the flight path throughout the flight from a ground station via a modem using a point-to-multipoint setting. Vehicles are not able to exchange information. Safety personnel on the ground stand by to take control with RC controllers (one person per vehicle) [8].

- 2014: Ten quadcopter UAVs in a decentralized swarm. Each vehicle is responsible for collision avoidance calculated with GPS data shared between vehicles via Xbee wireless radios. These position reports are sent in a broadcast mode between relative neighbors without establishing an end-to-end connection. Flight status information is sent to the ground station for monitoring and record keeping. The vehicles receive no information from the ground station pertaining to flight or organization. One individual can safely run the swarm due to the high level of autonomy [9].

- Ongoing Research: The swarming micro air vehicle network (SMAVNET) uses up to ten small fixed-wing UAVs (420g, 80cm wingspan) to conduct swarming behavior and communication experiments. Collision avoidance is accomplished via vehicle-to-vehicle communications while in flight. Neither operator-provided information nor on-board sensors are used to detect the in-air location of other vehicles. The SMAVNET team uses commercially available wifi hardware on their vehicles. Swarming behaviors are either reverse-engineered controllers or modeled after biological examples such as ants leaving pheromone trails [10].

Swarms have moved beyond individual vehicle control and tasking. As vehicles take more of the responsibility for their flight and mission completion, humans will be able to coordinate larger groups. Intel controls its 300 vehicle swarm on a single computer [11]. The vehicles are all pre-programed with their choreography. This allows the vehicles to execute their flight plan without communicating with other vehicles. The pre-flight work is substantial

but this differs from the NPS swarm, which is self-organizing and does not pre-plan collision avoidance through path planning but rather altitude assignments for each vehicle. With scale comes the ability to cover a larger area of operation. Vehicles are limited by their power, both in flight time and transmission range. Not all vehicles in a large group will have the ability to communicate with the ground station. One solution is to use multi-hop routing.

One potential solution to many issues described is the use of a Mobile ad hoc network (MANET). MANETs are groups of mobile nodes with the capability to receive, route, and transmit network traffic. These nodes are free to move which creates a dynamic topology. Mobility comes at the cost of being dependent upon a limited energy source as well as relying on wireless connections resulting in bandwidth constraints. These routing protocols fall in three different categories; proactive (table based), reactive, and hybrid schemes [12]. Originally designed to bridge the vast distances in space, DTN routing protocols use a store-and-forward tactic to deliver bundles of data when an end-to-end connection between source and destination is not possible [13]. There are many kinds of DTNs. Their design and performance are dependent upon their environment. As the NPS swarm grows in scale, multi-hop routing protocol may be a good way to ensure communication between vehicles.

The NPS swarm's various messages have different timing requirements. Some, like the heartbeat message, may not require every message to reach a vehicle, as long as one does within a defined window. It is not vital that missed heartbeats be delivered once the next has been sent by the ground station. Others, like asynchronous messages carrying behavior commands, must reach their destination or be able to inform the ground station that it was unable to reach the destination or vehicles will not join the correct behavior. More than one protocol may be chosen to work within the swarm's communication network, each chosen for a message that uses that protocols strengths to the swarms advantage. There are many to choose from. The ns-3 simulation software only had a few available at the time of this study. Section 3.5 reviews protocols that represent various tactics used in network routing. Some are well established, like the four applied in this study (DSDV, OLSR, AODV, and Epidemic). Others are relatively new as more variation is needed for unique wireless networks such as flying ad hoc networks (FANETs).

2.3 Mobile Routing Protocols

The following protocols are already supported by the ns-3 modeling software. What follows is a short high-level description of how each algorithm conveys data, their strengths, and their weaknesses.

Destination-Sequenced Distance-Vector Routing (DSDV): Nodes maintain their own routing tables. They also share their routing tables with their neighbors. This way, any node in the network knows how to route packets intended for a destination. Routing tables are populated with known nodes and the hop count to reach that node. Using hop count, nodes can relay information via the shortest path. Sequence numbers for routing table updates help eliminate stale data. Broken links are detected and updated by nodes that used to be immediate neighbors. When they discovery that they cannot complete the delivery, they remove the destination from routing tables completely [14]. DSDV is a proactive routing protocol. It continuously updates nodes on the current state of the topology at predetermined intervals. The advantage is that any node that has packets to send can immediately populate the header with the address and a loop-free route. The disadvantage is that each routing table update creates overhead. In a network with a large number of nodes, this overhead is quite taxing on the finite bandwidth available. Also, in a highly dynamic mobile environment like aerial vehicles, the topology may change too fast for routing table updates to reach distant nodes resulting in incorrect paths in the headers.

Dynamic Source Routing Protocol (DSR): Nodes populate packet headers with known routes held within their cache. If the destination is not in the node's cache, a Route Request is broadcast to all immediate neighbors. This Route Request travels from neighbor to neighbor until it is received by the destination node. The Route Request contains all previous hops used to travel to the destination node. A Route Reply is addressed with the reverse path, thus informing the source node and updating its cache. Packets waiting to be sent sit in the Send Buffer. These packets have two limits. First is how long the source will wait before sending new Route Requests. The second is the limit for how long the packet sits in the Send Buffer with no Route Reply. This is buffer management for difficult or unreachable destinations. Finally, if the source uses an old route from its cache that is no longer valid, the node that can no longer execute the route will generate a Route Error message. The source will then send a Route Request in order to find the updated route [15]. DSR is a reactive routing protocol. Nodes maintain their own cache. Route discovery is

only initiated when there is a specific need for one. The advantage is that overhead is limited to only those instances when it is absolutely necessary. The disadvantage is that the source must wait for route discovery when sending to destinations not listed in the cache. This wait is doubled when the cached route is old. This could be an issue for a highly mobile topology. [reference] states that this protocol is designed for highly mobile rates but also suggests an upper limit of about two hundred nodes.

Ad hoc On-Demand Distance Vector (AODV): Nodes broadcast Route Requests to find destinations they do not yet know. Route Replies are returned by either the destination or an intermediary node with a fresh enough route from itself to the destination. Freshness is determined by comparing the sequence number on the Route Request to the sequence number associated with the route held by the intermediary node. If the route sequence number is higher than the Route Request's, the intermediary node may send the Route Reply using its cached route to complete the routing node sequence. Nodes monitor their links to neighbors. If the link should break, a proactive error message is sent to other nodes that might want to use that route. Nodes use their precursor list, which is populated by Route Requests passing through or routed to them, to understand which nodes are affected [16]. AODV is a reactive routing protocol. Nodes maintain their own precursor lists as Route Requests pass through them or arrive. Route discovery is only initiated when there is a specific need for one. The advantage is that overhead is limited to only those instances when it is absolutely necessary. Link breaks are reported as they happen which reduces error messages. The disadvantage is that the source must wait for route discovery when sending to destinations not listed in their precursor list. This wait is doubled when the cached route is old but not for broken links. The overhead generated by proactive broken link updates may outweigh addressing delay.

Optimized Link State Routing (OLSR): This protocol reduces the high overhead of pure flooding by designating certain nodes as multipoint relays. Only these relays forward control messages throughout the network. Control messages contain information about the multipoint relay nodes: which other nodes they can reach and their hop counts. Forwarding nodes read these control messages and update their routing tables as they pass them along. OLSR is therefore a proactive routing protocol. Tables are updated via the regularly scheduled control messages [17]. Messages always have access to the shortest route at their destination and overhead costs typically associated with proactive routing are reduced

12

through the multipoint relay node designations. The disadvantage is that topology changes must wait for the next scheduled control message to be sent throughout the network. While waiting, a message may be sent along a path that no longer exists. In a highly dynamic topology composed of relatively fast-moving nodes, the network will either suffer from these erroneous path errors or increase overhead in order to update topology faster.

The following protocols are not currently in ns-3's libraries.

Epidemic Routing: MANETs assume a path from source to destination exists in an ad hoc network, even if it might change from moment to moment. DTNs do not require and end-to-end path at all. Though ns-3 does not contain any DTN protocols in its library, one DTN, Epidemic Routing, was implemented by [18].

Application-layer messages are distributed on nodes referred to as Carriers. The Carriers within the portion of the network connected to the source accept the message into their buffer space. As these mobile nodes travel, they may encounter another portion of the network not connected to the source. The message is copied to the buffers of Carriers in these encountered portions in order to increase the chance that the message is delivered to its destination. Mobile nodes frequently have limited resources and so, it is necessary to install limits on the number of hops a message can take before it is deemed an improbable delivery and dropped. Node buffer space is also limited and must therefore have rules to manage old and new messages carried in the buffer [19]. Epidemic permits the delivery of messages to otherwise unconnected elements of an ad hoc network but can be taxing on resources if not properly managed to suit the parameters of the network.

AeroTP: This protocol establishes an end-to-end connection much like TCP. Instead of one connection type, though, there are five levels of quality-of-service that one can use for packet transmission. They range from completely reliable (like TCP) to blindly sending with no guarantees (like UDP) [6]. Future work from these authors is to implement this protocol in the ns-3 simulator as well as for real world use. It seems like it would be beneficial to the NPS swarm if one could choose the level of quality-of-service required for different messages and in different environments.

MaxProp: A DTN strategy for routing messages in ground vehicle networks. As a node travels the network, it meets and creates a temporary link with encountered nodes. These

13

links are given a weight that indicates how likely the encountered node is to deliver a message to its destination. Node encounters are designed to transfer routing information, messages-in-buffer information, and actual messages [20]. MaxProp has been implemented in vehicles traveling street between five destinations with success, according to [20]. The NPS swarm adds the z-axis to the problem set as well as relative speed of nodes. Aerial vehicles are not limited to street pathways but intra-sub-swarm meetings can be predictable when in a holding pattern.

Probability Routing Protocol using History of Encounters and Transitivity (Prophet) v2: A DTN strategy that leverages patterns in node movement to predict the most likely nodes that will deliver a message to its destination. Probability thresholds can be set so that only the nodes most likely to successfully deliver a message will carry a copy of it [21]. Prophet v2 seeks to address the issue that Epidemic's flooding tactic can stress limited resources like node buffer space, bandwidth, and power.

Geolocation Assisted Predictive Routing (GAPR): A DTN strategy that leverages knowledge of other nodes' geolocation information in order to reduce the number of copies of a message traveling through a network [22]. The authors tested GAPR in a vehicle-based simulation using the ONE simulator. Buffer management was improved over other DTN like Epidemic, Prophet, and MaxProp, though it has yet to be tested in an aerial setting with all three dimensions and highly dynamic topology potential.

Zone Routing Protocol (ZRP): A hybrid routing protocol that uses zones to harness the strengths and reduce the weaknesses of proactive and reactive MANET protocols. The network is divided into zones. Within a zone, ZRP uses proactive routing and messages that must travel to a different zone use reactive routing [23]. The idea is that nodes in close proximity will exchange more messages than they will with nodes far away. This could be applied to sub-swarms.

2.4 Chapter Conclusion

There are many routing protocols with various strengths and weaknesses. The NPS swarm is unique from most of the other swarm research entities. One cannot try all protocols on the NPS swarm, but with the custom simulator developed through the work of this

thesis, one can narrow down the field of viable strategies. The next chapter will review the methods used to develop the simulator, the first routing protocols applied to this unique swarm environment, and the metrics used to measure the success of each protocol strategy.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Methodology

Chapter 2 discussed communication and routing challenges faced by multi-vehicle swarms as well as some of the protocols developed to address them. One reason there are so many protocols, many with variations tweaked by researchers, is that each environment and each swarm situation has unique features. It can be challenging to conduct real-world testing for the many options available. The creation of a simulator will allow many different trials to be run with no risk to hardware, minimal personnel involved, and reduced time investments. This chapter provides a review of the simulation built to model the NPS swarm in ns-3.

3.1 Simulation Program

Ns-3 is a discrete-event network simulator designed by the members of the NS-3 Consortium [24]. It contains many helpful tools within its library that replicate the behavior of real-world networks. These pre-coded tools can be pieced together a bit like one shopping in an electronics store for routers, antenna, and other network hardware. Other tools are coded to recreate environmental situations and stressors one expects the hardware to encounter. Though the NPS swarm uses commercial off-the-shelf hardware, the application-layer message generation and processing are unique. Ns-3 provides the flexibility to mirror the swarm's communication behaviors use of standard hardware while allowing for deviations from typical execution methodology.

3.2 Traffic Generation

Individual NPS swarm messages each have unique characteristics affecting behavior at the link layer that must be captured in the model. Synchronous messages described in Section 1.1 are broadcast at message-specific frequencies and have standard sizes. Additional messages of various sizes are broadcast asynchronously as required. Using the OnOffHelper in ns-3, setting the correct DataRate attribute and PacketSize attribute can produce the correct frequency, as illustrated in Table 3.1. Within the simulation, each message is also assigned a unique port over which to be received. All messages, regardless of type, include

a standard 16 Byte header added at the application layer. Each message application included a small addition or subtraction to start times. For the purposes of this simulation, it was important to ensure that applications were started at staggered times in order to ensure that collisions were not generated by poor coding practices. Asynchronous messages are sent infrequently enough that they do not contribute significantly to the stress induced by message volume. The code for asynchronous messages was developed but was not used while testing the different routing protocols as described in Chapter 4.

Table 3.1: Message Parameters

Message	Data Rate	Packet Size	Frequency
Heartbeat	160 bps	20 Bytes	1 Hz
Flight Status	768 bps	48 Bytes	2 Hz
Pose	4480 bps	56 Bytes	10 Hz
Red Pose	3200 bps	40 Bytes	10 Hz
Asynchronous	as required	variable	as required

3.3   Node Mobility

At the beginning of the code, the number of nodes is held by the variable nWifi. nWifi accounts for every mobile node and the stationary base station (APnode). Ns-3 has a feature called Container, which allows manipulation of multiple entities held within a single container through one reference. For example, one NodeContainer holds all vehicle nodes (wifiStaNodes) but not the base station node (APnode), which makes it easier to assign an action or attribute that will affect only the entities that will be in motion during the simulation.

To say the wifiStaNodes will be in motion is actually a bit misleading. The mobility model installed on the wifiStaNodes is the ConstantPositionMobilityModel. Instead of using one of the mobility models in ns-3, the wifiStaNodes can be moved exactly as they did during an actual flight. The NPS group can fly the swarm in a red versus blue scenario in which two competing groups attempt to target and engage all vehicles in the opposing swarm [2]. The vehicles are assigned a swarm ID, which places them in either sub-swarm one (red) or sub-swarm two (blue). After launch, each vehicle flies to its assigned altitude and begins circling a designated area. Once all vehicles are launched, there will be two columns of circling

UAVs standing by for their next order, which will arrive via an asynchronous message. The two sub-swarms will come together and simulate an air-to-air engagement, then separate back into their designated columns. After all behavior scenarios are complete, individual UAVs will be told to land until all vehicles are safely on the ground. The illustrative figures from [2] is included as Figure 3.1 for reference. Table 3.2 illustrates the composition of the four swarm events used for the mobility model in this thesis.
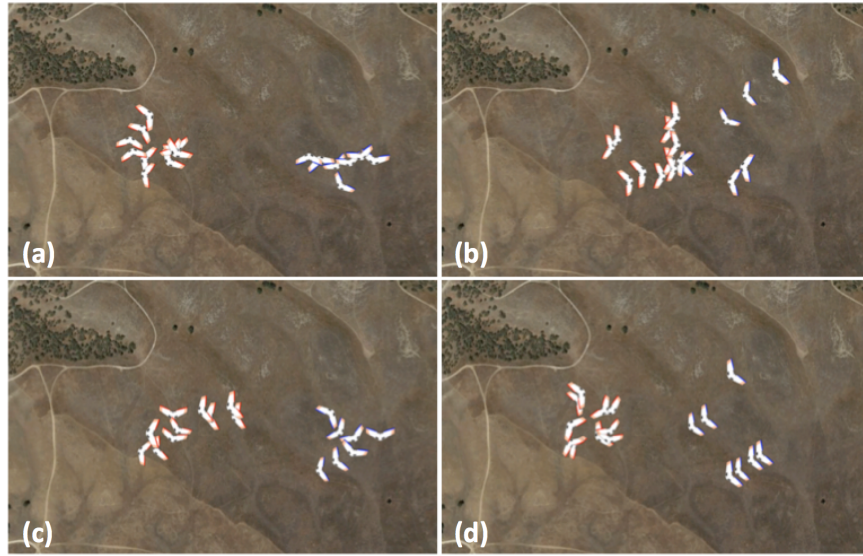

Figure 3.1: Visualization in Google Earth of live-fly field experiments of 10v10 flights in December 2015. (a) Red swarm (left) and Blue swarm (right) in `swarm-ready` state; (b) Red and Blue swarms engaging using Naive Shooter algorithms; (c) Swarms disengaging for reset; (d) Blue swarm egressing at conclusion of experiment. Source: [2]

Once all the elements of the swarm's communication architecture were coded, a trace file for a twelve-vehicle flight was used as the mobility model in order to ensure the model could produce results that mirrored the performance of an actual swarm. The section labeled EXTERNAL TRACE reads each line of the .csv file from the NPS swarm log and places the elements in their appropriate variables. Each line has six fields: GPS-reported time, swarm ID, node ID, x-coordinate, y-coordinate, and z-coordinate. The EXTERNAL TRACE section adjusts the first time variable to start at 0 seconds within the simulation and reflects that relative change in all successive time variables. It is important to ensure the correct number of nodes (plus one for the APnode) is set, the simulation is allotted enough time to cover the full duration of logged mission time, and that any other node mobility

modules are commented out before running.

Table 3.2: Swarm Configurations

Mobility Model	Ground Station	Sub-swarms	Sub-swarm nodes	Simulation nodes
6 v 6	1	2	6	12
10 v 10	1	2	10	20
15 v 15	1	2	15	30
25 v 25	1	2	25	50

3.4   Analytical Code

Ns-3 produces a data file at the conclusion of any run in the form of an American Standard Code for Information Interchange (ASCII) trace file (.tr). A stand-alone program was written to read the .tr file and create a readable table of the data points needed for performance analysis. Python v3 was the chosen language due to its helpful list sorting libraries. The tables produced after processing the trace file provide total bytes received by each node, the time over which each node received packets, a count of how many times one node sent information intended for another node or nodes, a count of how many messages were received by each node (that were addressed to that node), and how many times each node helped forward messages on to the final destination.

As discussed previously, the NPS swarm relays messages by broadcasting them to all nodes. Once a message is received by a node, the application layer checks to be sure the destination address matches that node's ID and messages that are not intended for that node are dropped. These instances were filtered out of the count for messages received since their safe arrival does not help in the mission of the swarm. It is more important to understand how many intended messages were received since their presence or absence directly affects the behavior of the vehicle.

The trace file contains more lines than simply those illustrating messages sent and received between nodes. The intention of the data analysis was to understand the volume of swarm function related messages sent and successfully received so messages that served a purely network layer function were not counted in the final message tallies. For example, internet control message protocol (ICMP) error messages were not counted. Nor were the plethora of address resolution protocol (ARP) messages that traverse the network at the beginning

of a simulation. Since these messages were creating too many collisions, a function that completes the job of ARP messages outside of normal traffic flow was added in order to see simulator results more purely based on swarm message movement.

Once the simulator was producing acceptable numbers on a small scale, routing protocols were added to the testing. These protocols will be addressed in section 3.5. Proactive and reactive protocols produce their own messages in order to discover and update multi-hop routes in the network. Some of these messages began polluting the sent and received messages tallies and so, filters for these were added. Finally, the Epidemic protocol generates multiple copies of a message. A filter was applied to ensure only one instance of a message was counted as received while duplicate messages arriving at the destination were not.

3.5 Routing Protocols

Once the simulator reproduced message traffic in a broadcast configuration, code was added for additional routing protocols that are representative of the major categories traditionally used in ad hoc networking. The ns-3 library provides a number of useful ad hoc routing protocols: OLSR, a link state proactive protocol, DSDV, a distance vector proactive protocol, and AODV, a reactive protocol. In addition, a simple flooding DTN, Epidemic, was implemented in ns-3 by [18] and was available for addition to the simulation as well. The details of how these protocols work are outlined in Chapter 2. Though these protocols are not optimal for use in FANETs, they can be treated a representatives of fundamental routing tactics and are therefore useful in demonstrating strengths and weakness inherent to all protocols that fall within these respective categories.

- OLSR: a proactive protocol is able to maintain tables for immediate addressing of the shortest path. For cases when fewer nodes are within range of the ground station, OLSR should be able to quickly find a multi-hop pathway for the message. On the other hand, when the ground station is able to reach most or all nodes directly, link state messages will likely create overhead for unnecessary multi-hop path tables. In this case, the benefit of table maintenance will be lost and limited bandwidth resources will be used with no returned value. Also, as the number of nodes increases, OLSR will have a more difficult time converging and the highly dynamic topology will cause the generation of a large number of link state messages.

- DSDV: carries the same pros and cons as a proactive protocol. DSDV should perform better than OLSR, though, because it focuses more on updating the local neighborhood than on global convergence. This, in turn, should result in fewer distance vector messages. The proactive nature of this protocol will still generate extra messages for table updates in a network even if all nodes are in single hop range of the ground station, thus stressing bandwidth unnecessarily.

- AODV: a reactive protocol should have lower overhead costs when compared to proactive protocols. These overhead cost savings should be most evident when more nodes are within single hop range of the ground station, and route discovery will not be invoked as often.

- Epidemic: a DTN should perform best when fewer nodes are within single hop range of the ground station. Because DTNs do not require the existence of an end-to-end path to exist, there should be no latency due to route discovery or slow network convergence and no route errors. Epidemic performance will, however begin to degrade as the number of nodes increases due to the number of messages traversing the system.

The YansWifiPhyHelper has the option to set transmit and receive gains above the default setting. In order to simulate varying levels of node connectivity to the ground station, each protocol was run under six different gain settings: 0, 3, 6, 9, and 20 decibels. Transmit and receive gains were always set to matching levels. The 0 gain simulated many nodes outside the range of the base station and required routing protocols to utilize their multi-hop abilities in order to help deliver messages. Increasing gain, on the other hand, results in more single hop routing and less reliance on the routing protocol for delivery. With a gain setting of 20, all nodes should be within single hop range, and the impact of running routing protocols with no multi-hop benefit will be evident. Once basic simulator functionality was confirmed, experiments were conducted swarm trace data for twelve, twenty, thirty, and fifty vehicle events.

3.6   Metrics Used

The following metrics were utilized for quantitative comparison of protocol performance:

Goodput: calculated by dividing the total number of bytes by the time taken to receive those bytes. This is a good indicator of how much useful information was able to reach the intended node. Average goodput ignores the presence of retransmissions, duplicates, or forwarded messages in the network and focuses instead on the amount of useful data a node receives during the simulation.

Packet Delivery Ratio (PDR): calculated by dividing the number of messages received by the number sent. This provides a percentage of the messages that made it to their destinations. The ability for a network to deliver timely communications to and from nodes is key to synchronized swarm operation. This metric provides an indication of network configuration reliability.

Overhead and Collisions: observed in trace files. The data link layer retransmits messages that are not sent due to collisions meaning the number of sent messages can exceed the expectation. This is not the same as TCP retransmissions but has a similar effect. Every time a message must be retransmitted, latency suffers. High overhead generated by routing protocols, either though routing table updates, route discovery, or flooding, taxes bandwidth resources and causes collisions. Understanding the impact of this phenomenon is therefore an important component to assessing swarm network performance.

3.7   Chapter Conclusion

This chapter detailed the construction of the simulator using the various tools provided by the ns-3 libraries. It also detailed the mobility model derived from real world swarm testing. Results in Chapter 4 should be close to what occurs in the wireless environment used by the vehicles and their ground station. If this holds true in the data analysis, then results from the addition of the four new routing protocols should follow the assumptions outlined in section 3.5 and therefore be a good predictor of how other protocols in each family should affect swarm communications.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Results

Chapter 3 covered the details of experimentation. It also broadly outlined the effects expected when adding routing protocols to the simulation and adjusting the gain to include a different portion of each swarm. Chapter 4 will review the results from the 120 simulations and highlight any illustrated trends and relationships. As is common in the realm of experimentation, reality is not quite what was expected but this does inform approaches one should take as the simulator advances to future work.

4.1   Average Goodput: Ground Station
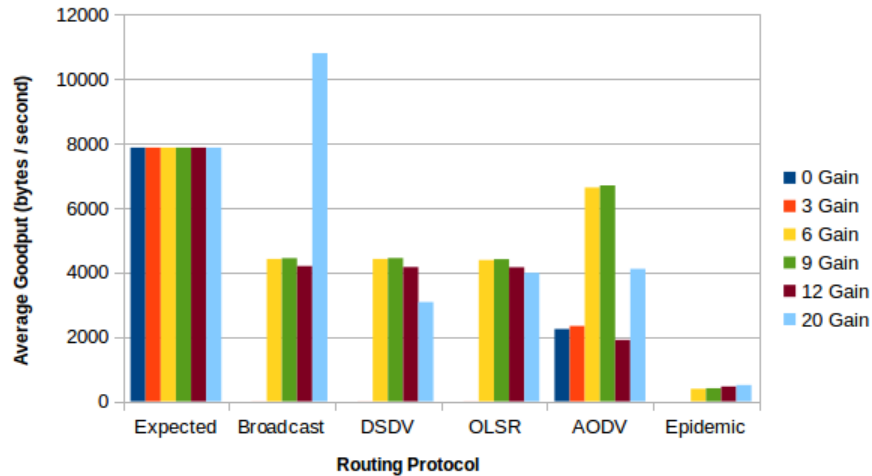
Figure 4.1: Average Goodput to Ground

6 v 6 Swarm

Performance metrics can be broken into two broad categories; messages intended for the ground station and messages intended for the vehicles or nodes. Ground station metrics are messages generated by the swarm of vehicles flowing down to a central point, which consists of the flight status and pose messages. As illustrated in Table 3.1 in Chapter 3, flight status messages are 48 bytes in size and each UAV sends two every second. Pose messages are 56 bytes in size and each UAV sends ten every second. In a perfect networking

environment, each vehicle is sending 12 messages, for a total of 656 bytes, to the ground station each second.

Figure 4.1 shows the average goodput of the 6 v 6 swarm while using the different routing protocols and across the varied levels of gain. DSDV and OLSR perform at about the same level as simple UDP broadcasting except for when all vehicles are within single-hop range of the ground station. Broadcasting no longer loses messages to being out of range while DSDV and OLSR, the two proactive protocols, experience about a 50% loss, likely due to collisions caused by routing table updating message traffic. (Collisions were observed in the trace files generated by the simulation, but were not counted.) The fact that broadcasting in the gain 20 environment delivers more bytes to the ground than had been sent is curious. AODV, the reactive protocol, performs more consistently and is even able to deliver messages to the ground in the environments where the gain is at its lowest settings. It also performs closer to the expected levels at gains 6 and 9. The DTN routing protocol, Epidemic, performed poorly in all gain environments. Goodput was much lower than any other protocol, though it produced a high volume of link layer messages, which was observed in the trace files. The files for this protocol were larger than any other due to routing and message duplication. In fact, the simulator would not always finish the Epidemic simulations. The 6 v 6 swarm has the only complete data set and therefore Epidemic results have been dropped from the remaining swarm data with a comfortable assumption that performance would only worsen as more vehicles are added to the swarm.
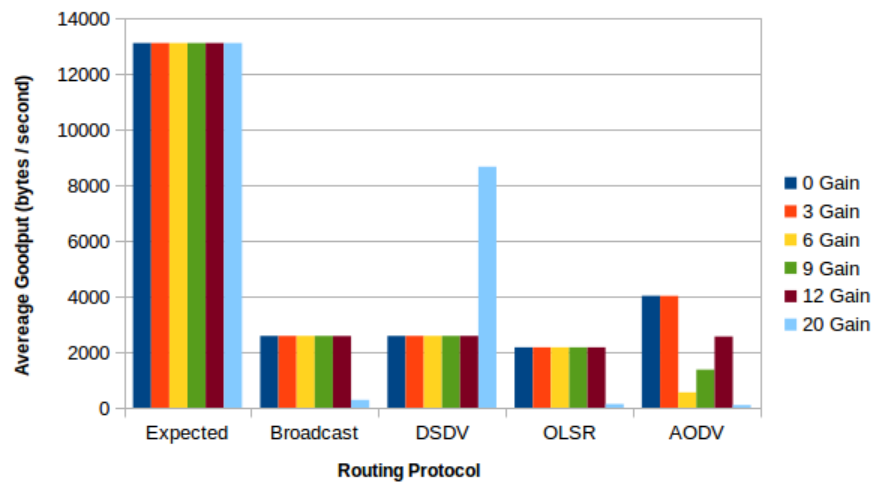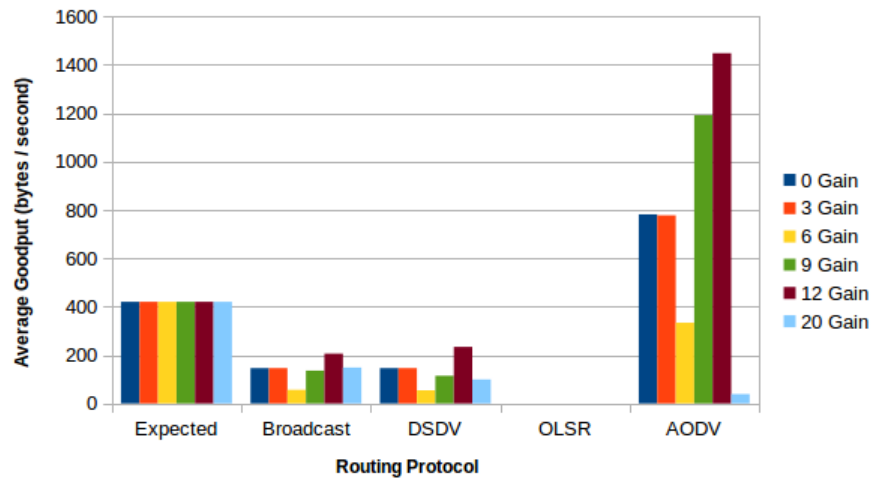
Figure 4.2: Average Goodput to Ground

10 v 10 Swarm
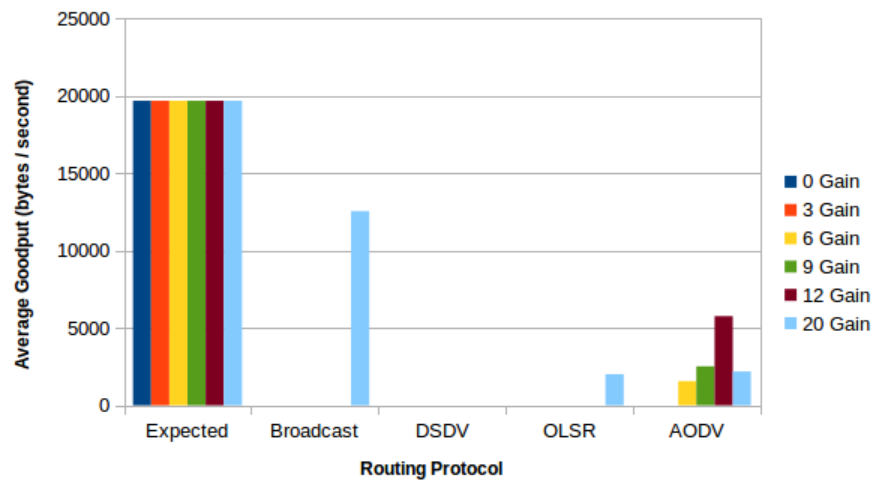
Figure 4.3: Average Goodput to Ground

15 v 15 Swarm

27
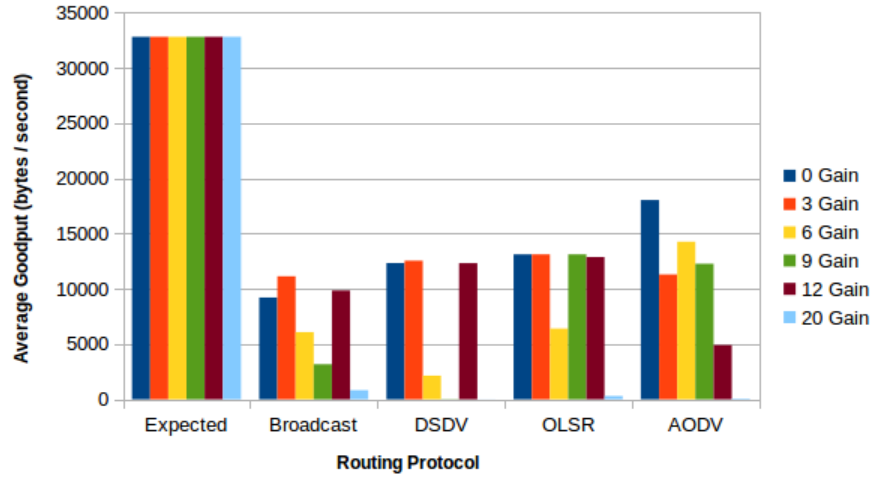
Figure 4.4: Average Goodput to Ground
25 v 25 Swarm

As the number of vehicles increases in the swarm of 10 v 10, illustrated in Figure 4.2, we see similar trends but instead of about a 50% loss of data delivered, this drops to 80%. Unlike in 6 v 6, broadcast drops its performance in the gain 20 environment while DSDV improves.

Figure 4.3 shows a severe decline in performance once the swarm size reaches 15 v 15 vehicles. AODV is able to deliver messages in most environments and broadcasting in the environment of gain 20 once again shows a strong rebound with a majority of vehicles in single-hop range.

In the largest swarm, Figure 4.4, we see a return to about 50% of expected values across the routing protocols with AODV and OLSR producing the most consistent goodput for ground metrics.

4.2   Average Goodput: Nodes
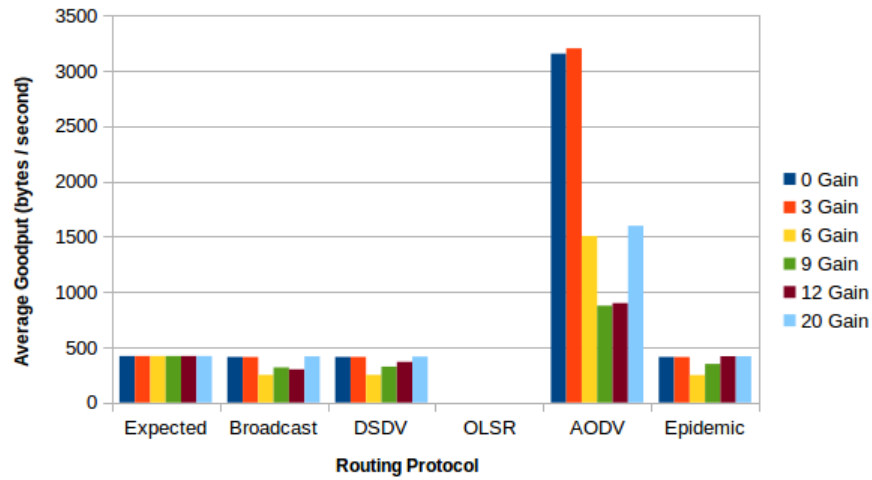
Figure 4.5: Average Goodput to Nodes

6 v 6 Swarm

Figure 4.6: Average Goodput to Nodes

10 v 10 Swarm
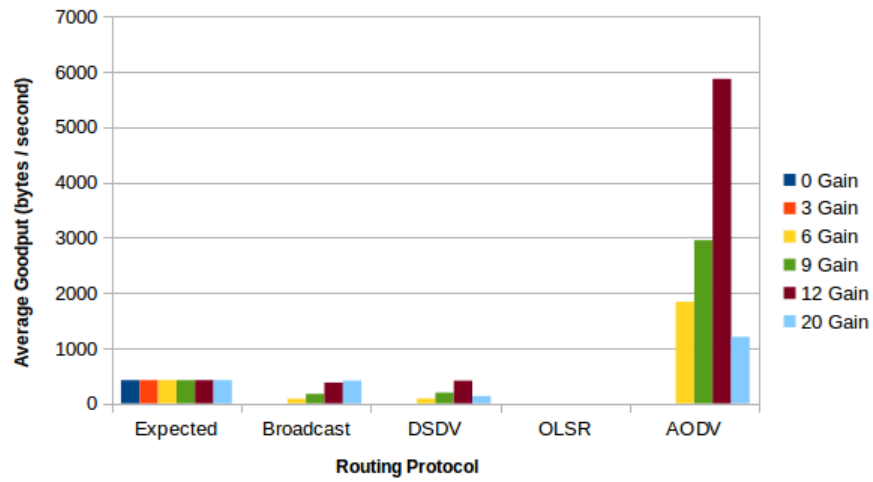
Figure 4.7: Average Goodput to Nodes
15 v 15 Swarm
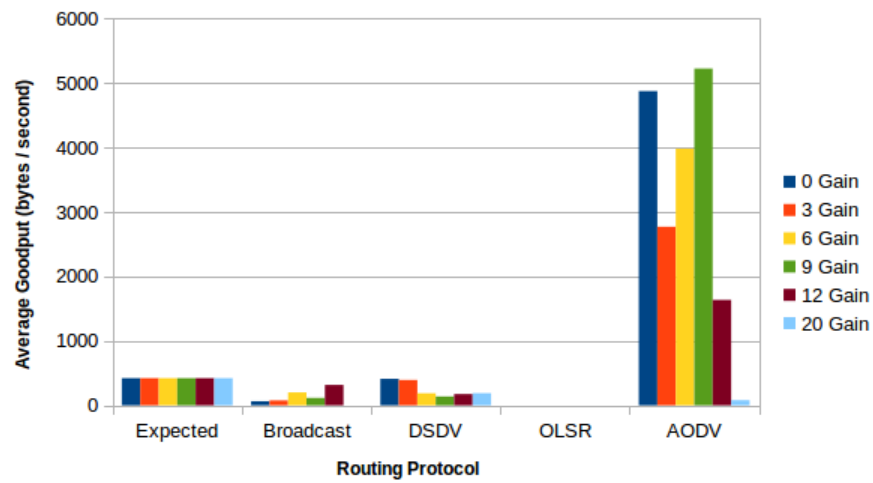
Figure 4.8: Average Goodput to Nodes
25 v 25 Swarm

30

Node metrics are messages generated by the ground station and sent to the broadcast address of 10.1.3.255, which every vehicle acknowledges as addressed to itself. This traffic is composed of the heartbeat and red pose messages. As illustrated in Table 3.1 in Chapter 3, heartbeat messages are 20 bytes in size and the ground station sends one to the collective group of UAVs every second. Red pose messages are 40 bytes in size and ten are broadcast every second. In a perfect networking environment, the ground station is sending 11 messages, for a total of 420 bytes, every second to each vehicle. Average goodput was calculated for each node and then those totals were averaged in order to obtain a general average goodput for all nodes in a swarm. This average if preferable to demonstrate the overall success of messages destined for vehicles rather than a graph that shows the goodput of every vehicle individually. That being said, some detail was lost in the graph but was observed in the data sheets. In the 10 v 10 swarm, not all vehicles received messages. The graph shows the average goodput for those vehicles that received data, which was nearly always half, affected by the 0 bytes received by the other half. Only broadcast and DSDV at gain 20 contained 13 and 15 vehicles respectively in the average goodput chart. Generally, the same vehicles did or did not receive messages across the varied gain environments in AODV and in the extremes for broadcast and DSDV (gains of 0, 3, and 20). The ten or so vehicles that did not always receive messages in this swarm configuration do not correspond with vehicles in a single sub-swarm. There was a mix from both.

Looking at Figures 4.5, 4.6, 4.7, and 4.8 one can see AODV far exceeds all protocols and even the expected average goodput while OLSR failed to deliver any information to the nodes. Observation of the trace files show messages transmitted by the ground to the nodes are OLSR's link layer messages. These are successfully received by the nodes, but the ground station must update again and again as the topology changes. It never transmits packets that contain swarm messages.

In Figures 4.9, 4.10, 4.11, and 4.12 AODV has been removed in order to better compare the expected average goodput with the simple broadcast and DSDV. It becomes clear that they once again behave similarly. They are most successful in the 6 v 6 swarm, meeting expected values in environments of gain 0, 3 and 20. Likewise, they all degrade or improve in a similar manner as the swarm vehicle numbers increase.
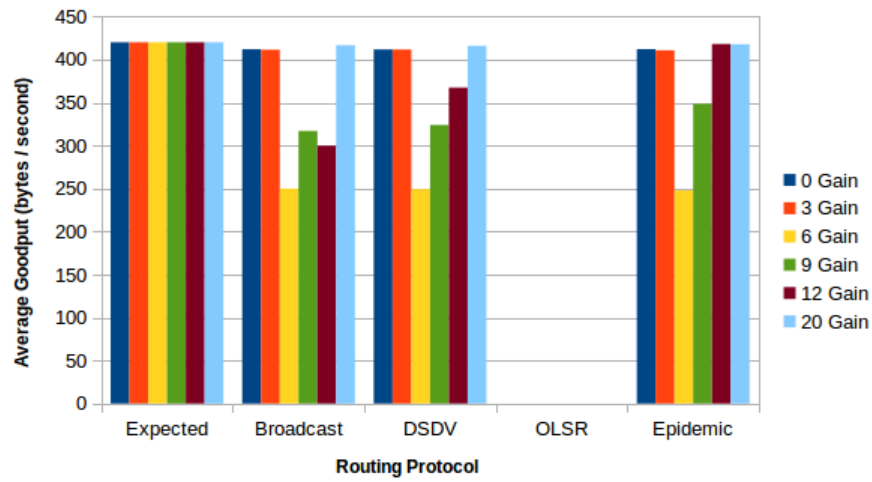
Figure 4.9: Average Goodput to Nodes (no AODV)
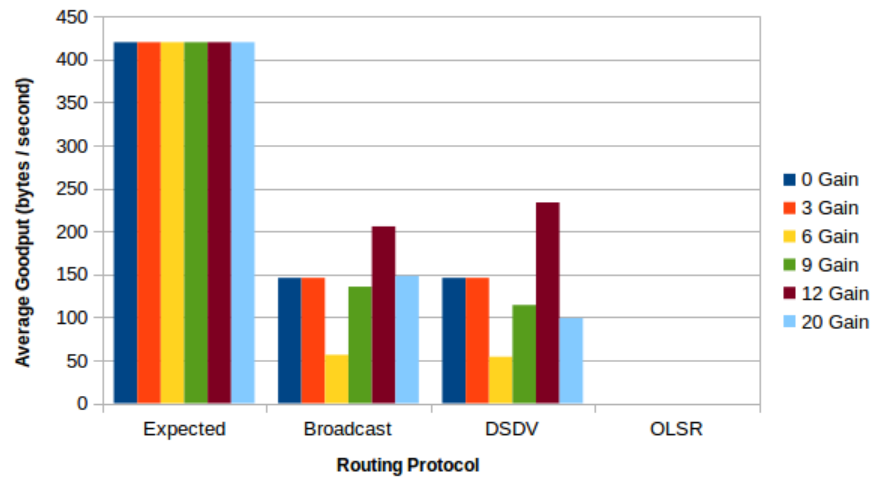
6 v 6 Swarm

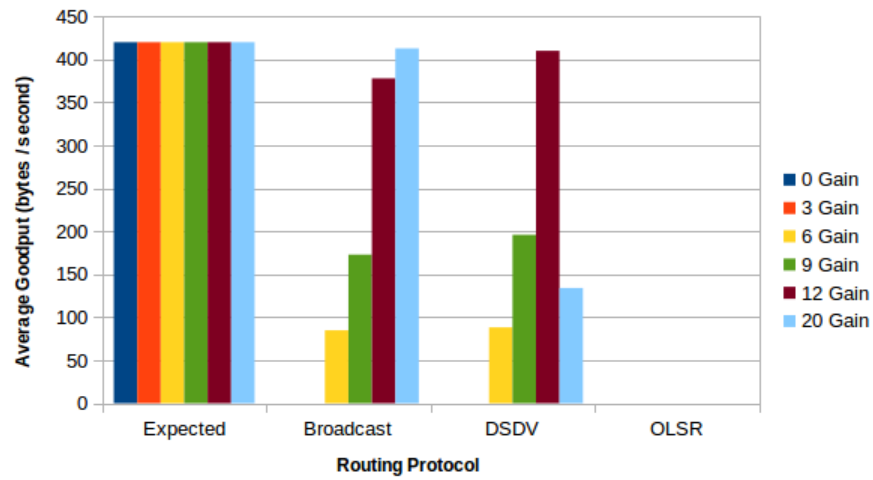Figure 4.10: Average Goodput to Nodes (no AODV)

10 v 10 Swarm

Figure 4.11: Average Goodput to Nodes (no AODV)
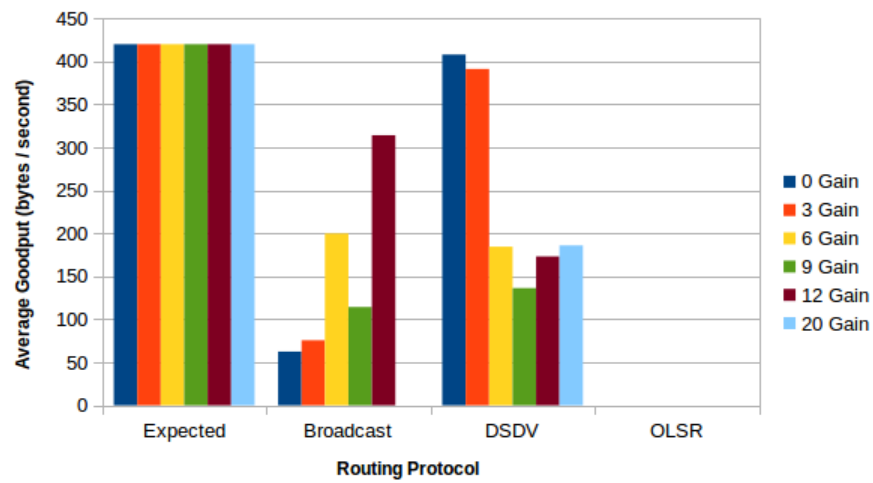
15 v 15 Swarm

Figure 4.12: Average Goodput to Nodes (no AODV)

25 v 25 Swarm

4.3 Packet Delivery Ratio

PDRs for flight and pose messages were calculated by dividing total messages received by total messages sent. Heartbeat and red pose messages are sent once from the ground station to the broadcast address. Each node counts a message received from the broadcast address which results in their messages received numbers multiplied by the number of vehicles in the swarm receiving them. To account for this, each total for messages received is divided by the number of swarm vehicles to get an average number one vehicle received. This adjusted number is divided by total messages sent. A similar adjustment is applied to the "all messages" average as well. The adjusted number of broadcast messages received by a node is added to the messages received by each node from the ground station and that total is used for the blue "all" PDR trend line in the figures that follow.
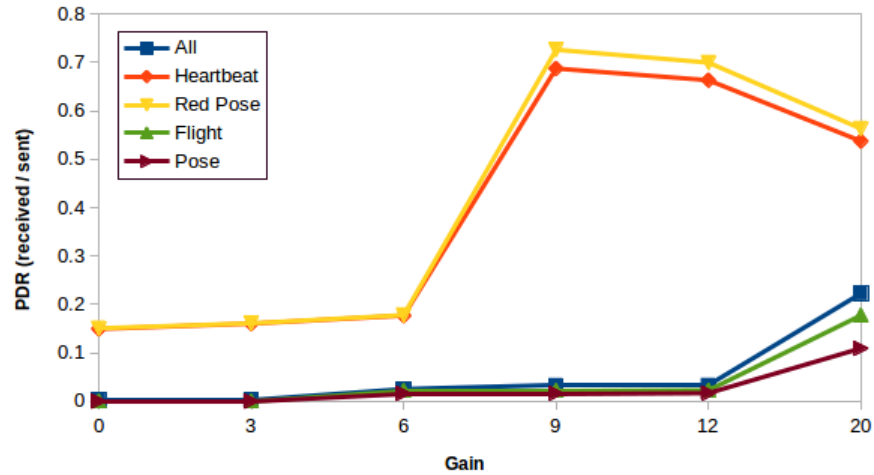
Figure 4.13: Packet Delivery Ratio Using Broadcast
6 v 6 Swarm
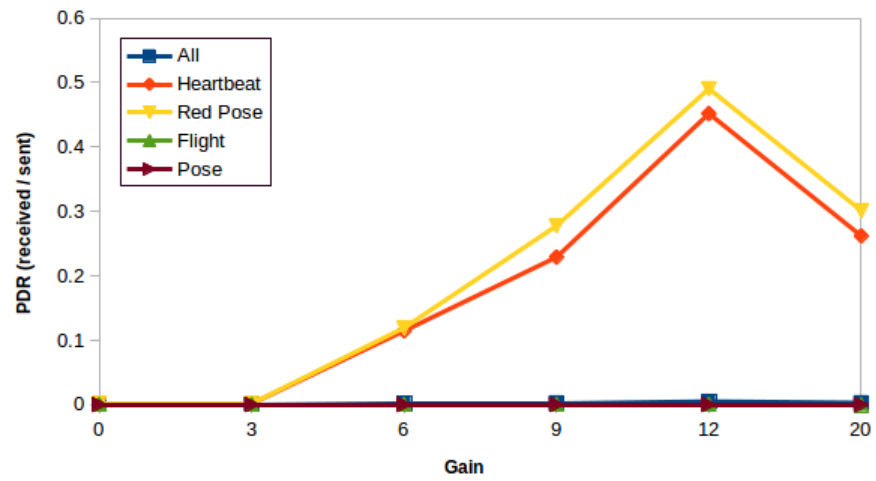
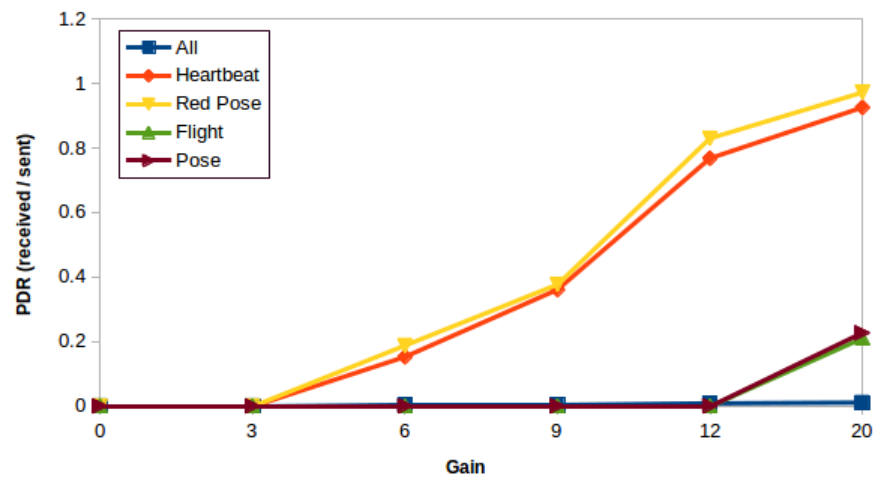Figure 4.14: Packet Delivery Ratio Using Broadcast
10 v 10 Swarm

Figure 4.15: Packet Delivery Ratio Using Broadcast
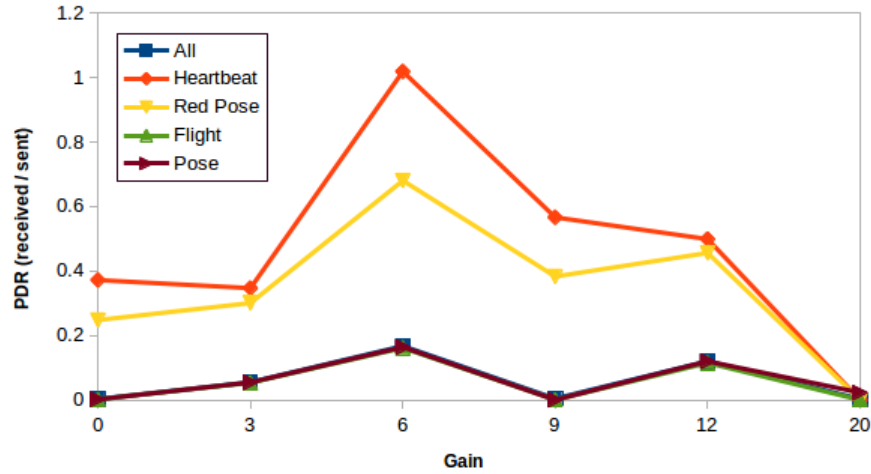15 v 15 Swarm

Figure 4.16: Packet Delivery Ratio Using Broadcast
25 v 25 Swarm

In general, the broadcast method improves as the gain in the environment increases, as shown in Figures 4.13, 4.14, 4.15, and 4.16. The exception is the 25 v 25 swarm, where performance decreases with gain after the gain 6 environment. It may be these environments have become saturated. Packets traveling from the vehicles to the ground achieved a higher packet delivery ratio than those traveling from the ground station to the vehicles. The close correlation between packet delivery rates for different message types traveling from the same sources is easy to observe. Heartbeat and red pose messages destined for the vehicles are close in values and patterns. Likewise, flight status and pose messages also behave in a close relationship. The blue line graphing total messages received follows the pattern of the flight status and pose messages as well. The expectation would be for this line to fall between the two trends. The success or failure of delivering packets to these two general destinations should pull the summary between the them especially because one set is relatively successful (vehicle-destined) and the other performs poorly (ground-destined).
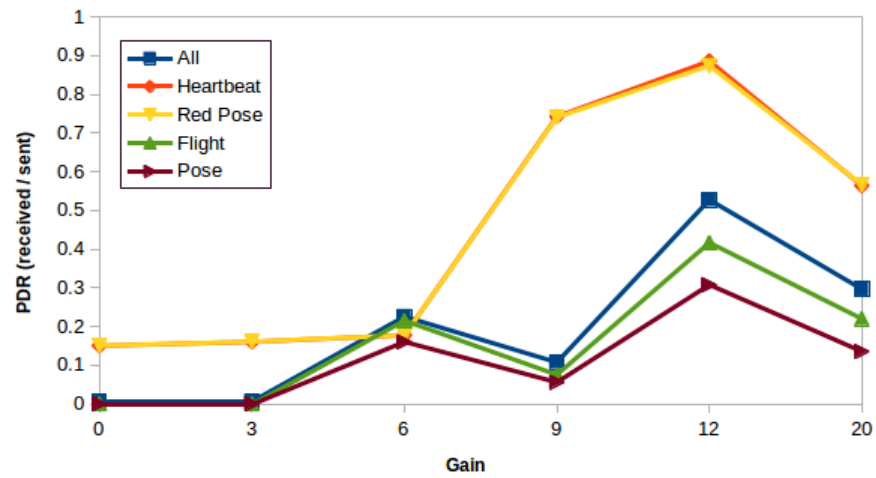
36

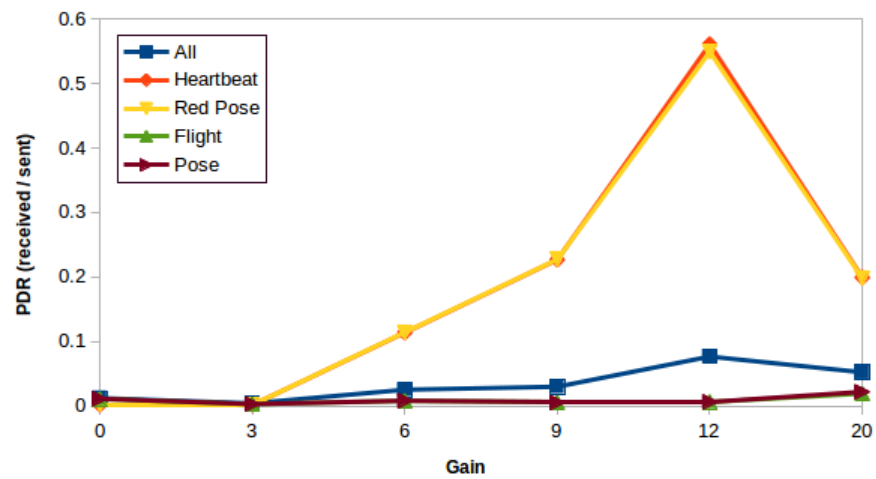Figure 4.17: Packet Delivery Ratio Using DSDV

6 v 6 Swarm

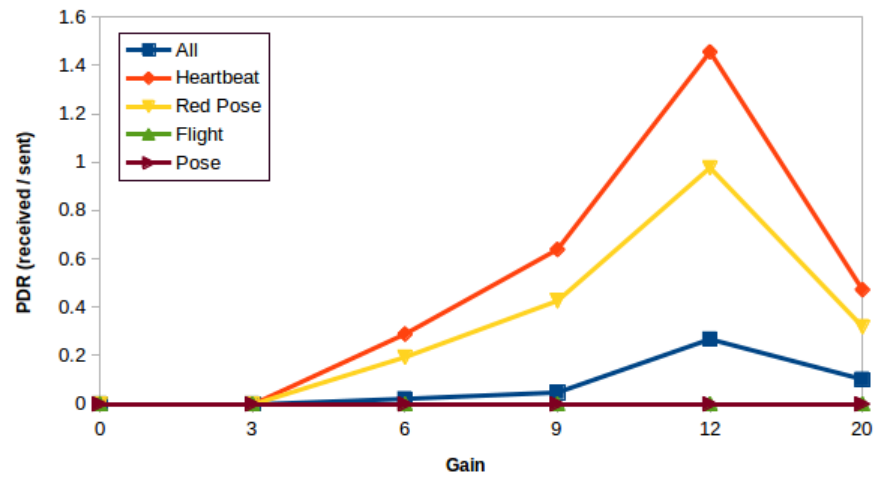Figure 4.18: Packet Delivery Ratio Using DSDV

10 v 10 Swarm

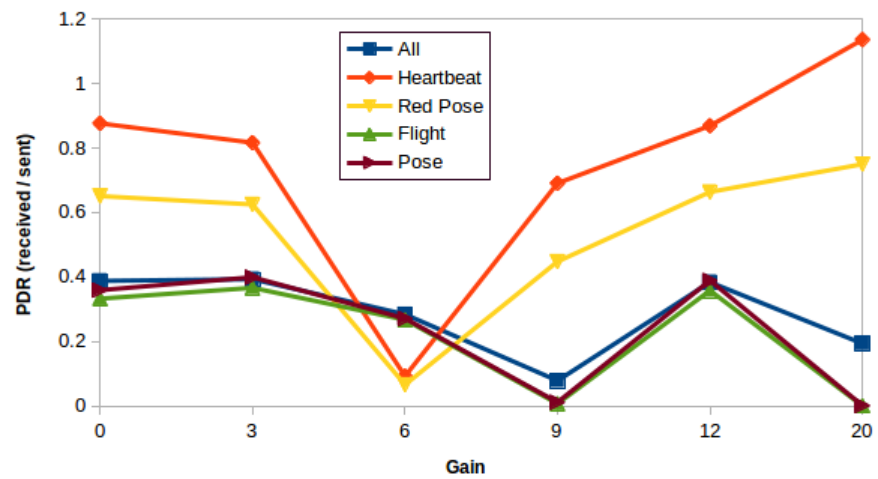Figure 4.19: Packet Delivery Ratio Using DSDV

15 v 15 Swarm

Figure 4.20: Packet Delivery Ratio Using DSDV

25 v 25 Swarm

DSDV's best performance was in the gain of 12 environment across all swarm sizes. (Figures 4.17 - 4.20) Curiously, though, heartbeat messages produced packet delivery ratios above 1.0 in the larger swarms. The blue line that indicates the total messages received does fall between the two trends in this set of data.

AODV's goodput performances are mirrored well in the packet delivery ratio. (Figures 4.21 - 4.24) Performance is best in the smaller swarm of 6 v 6 after reaching the gain of 6 environment. Delivery rates are low in the 10 v 10 and 15 v 15 swarms. The 25 v 25 swarm maintains around 50% of packets delivered, peaks in the gain of 12 environment and then crashes in the most inclusive gain environment. This crash is in contrast to the behavior in the 6 v 6 swarm, which performs closer to expectations. As more vehicles are within a single-hop distance to the destination (either ground or vehicles), AODV should not falter like DSDV and OLSR. This is due to its reactive nature. Overhead will decrease as multi-hop routing tables are less frequently required. Yet it was observed that the broadcast numbers also crashed in the gain of 20 environment in the 25 v 25 swarm where there was no routing. It can be inferred, then, that this crash was not caused by AODV routing protocol's on-demand routing messages trying to deliver messages to multi-hop destinations. It is more likely tied to the mobility model used.

OLSR's packet delivery rates reflect the poor performance illustrated in the average goodput bar graphs. (Figures 4.25 - 4.28) Rarely are more than half of sent packets received. The vehicle-destined heartbeat and red pose messages, which have thus far outperformed all other messages using the other routing protocols, lay at the bottom of these graphs. It observed in the trace files that OLSR has a difficult time creating the routing tables for vehicle destinations in such a dynamic topology. Looking at the raw trace files generated by the simulator, one can see numerous routing table messages. They far outnumber actual message traffic. Their frequency creates many collisions at the data link layer. This lower layer collision detection will only attempt a retransmission so many times before it gives up.
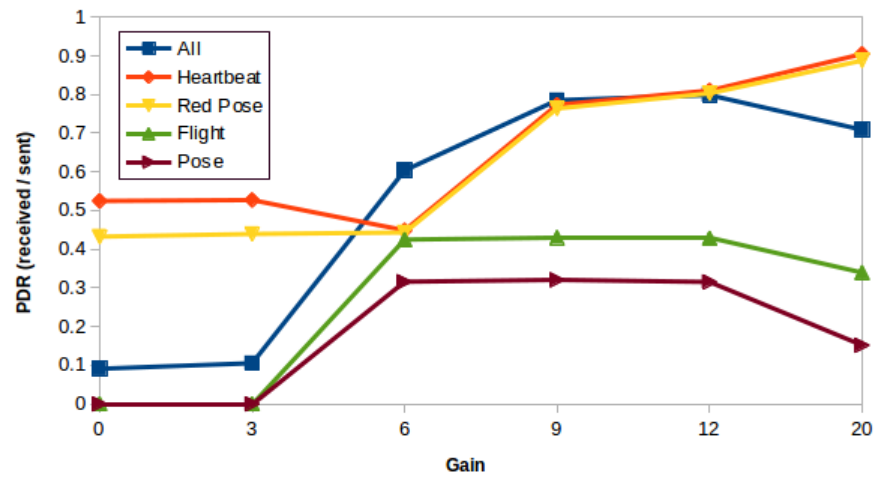
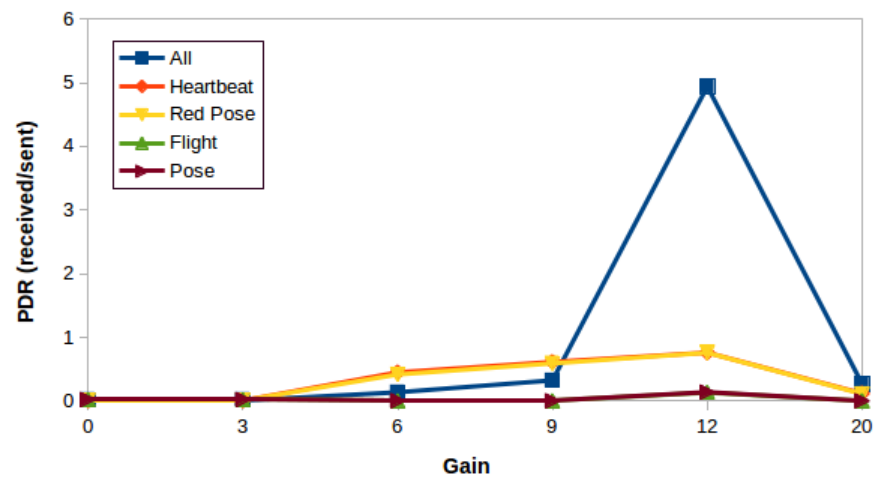Figure 4.21: Packet Delivery Ratio Using AODV

6 v 6 Swarm

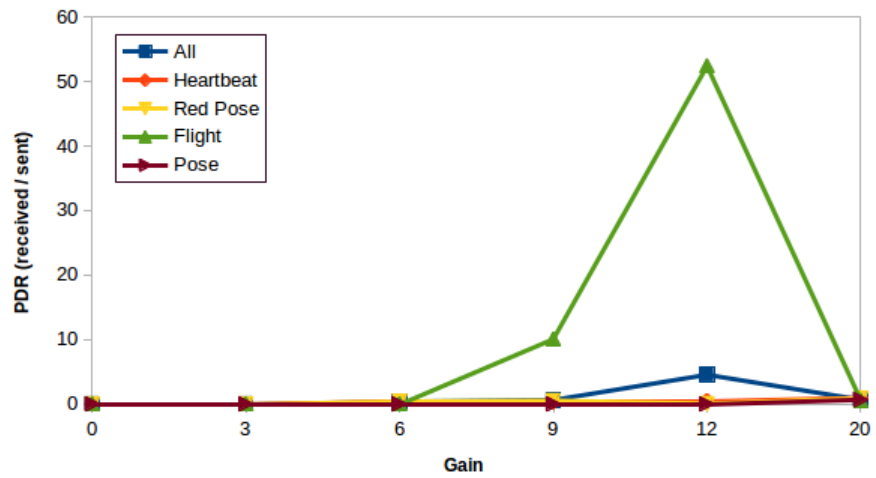Figure 4.22: Packet Delivery Ratio Using AODV

10 v 10 Swarm

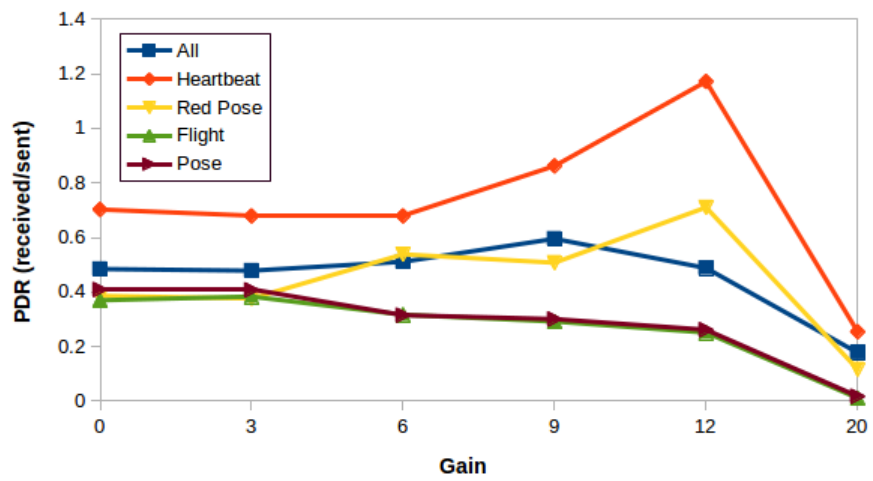Figure 4.23: Packet Delivery Ratio Using AODV

15 v 15 Swarm

Figure 4.24: Packet Delivery Ratio Using AODV
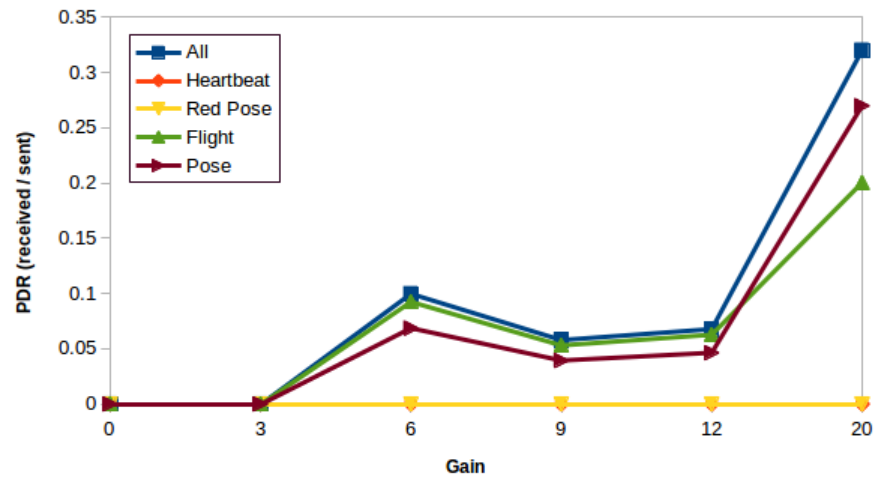
25 v 25 Swarm

Figure 4.25: Packet Delivery Ratio Using OLSR
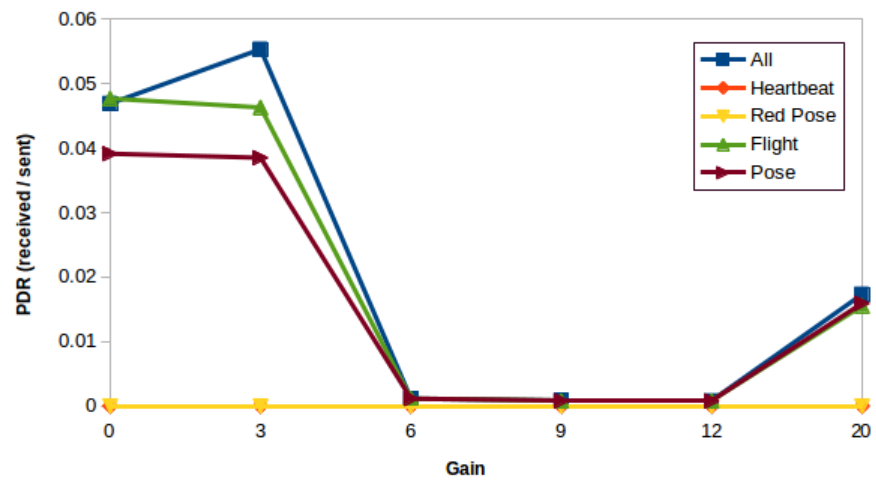
6 v 6 Swarm

Figure 4.26: Packet Delivery Ratio Using OLSR
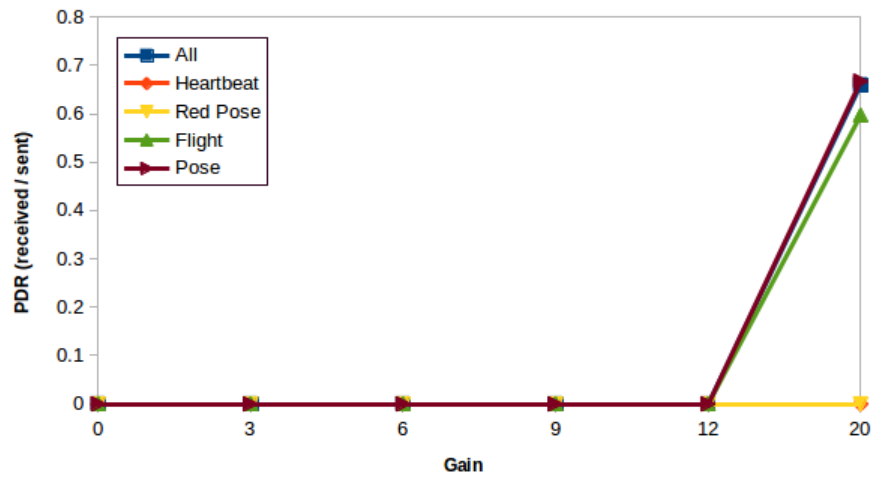
10 v 10 Swarm

Figure 4.27: Packet Delivery Ratio Using OLSR
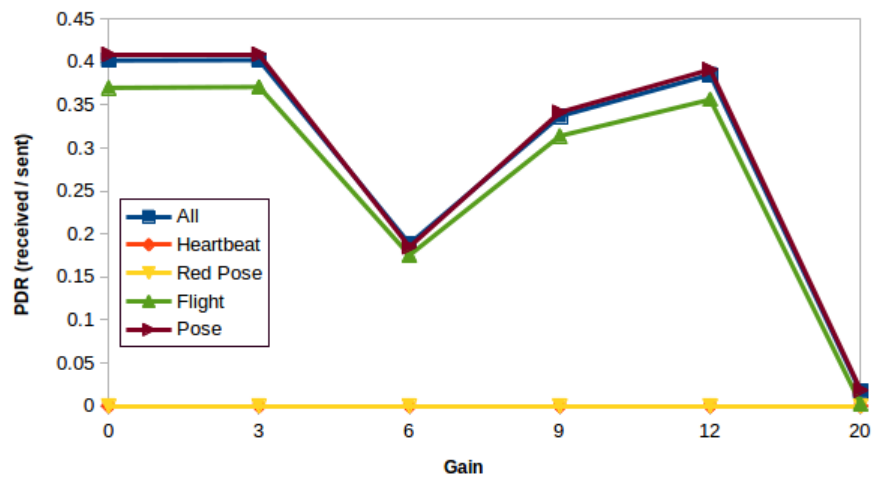
15 v 15 Swarm

Figure 4.28: Packet Delivery Ratio Using OLSR
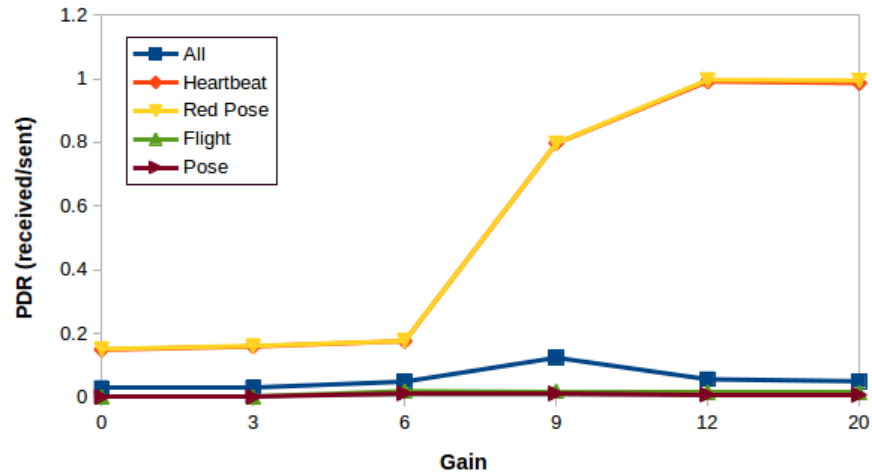
25 v 25 Swarm

Figure 4.29: Packet Delivery Ratio Using Epidemic
6 v 6 Swarm

Epidemic's PDR performance in the 6 v 6 swarm show that it can deliver messages to the nodes as the environment becomes more inclusive (more nodes able to be touched by the flooding) but the nodes are never really able to deliver messages to the ground. (Figure 4.29

Figures 4.30 - 4.33 gives a final perspective of PDR by placing the routing protocols' all messages side by side per swarm size. AODV PDR remains above them all in every swarm. The recovery of stats in the 25 v 25 swarm is even more evident. It is likely that mobility models influenced overall performance in greater magnitude than expected.
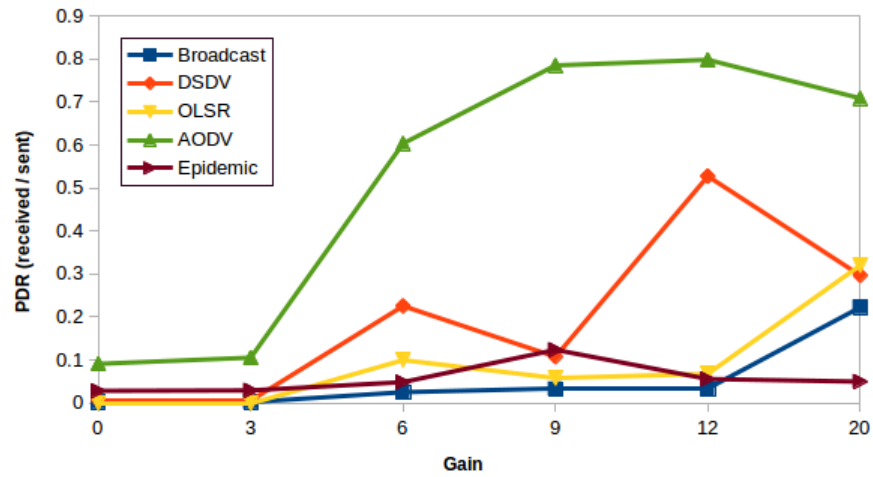
Figure 4.30: PDR of All Messages by Routing Protocol
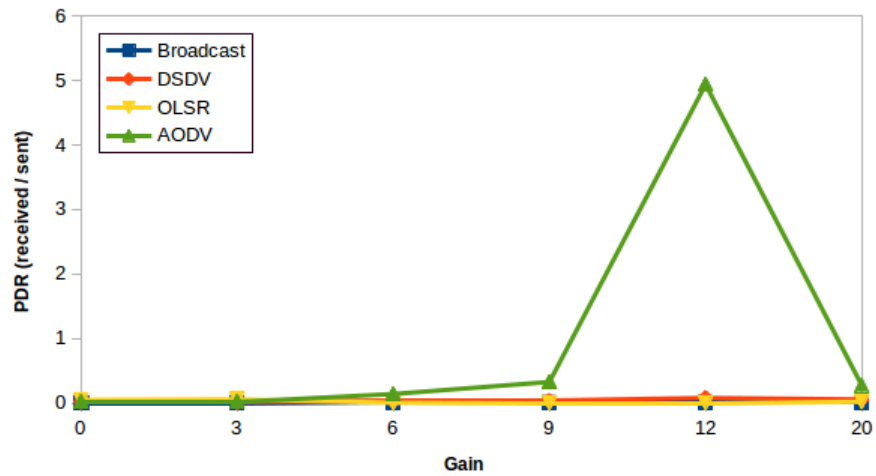
6 v 6 Swarm

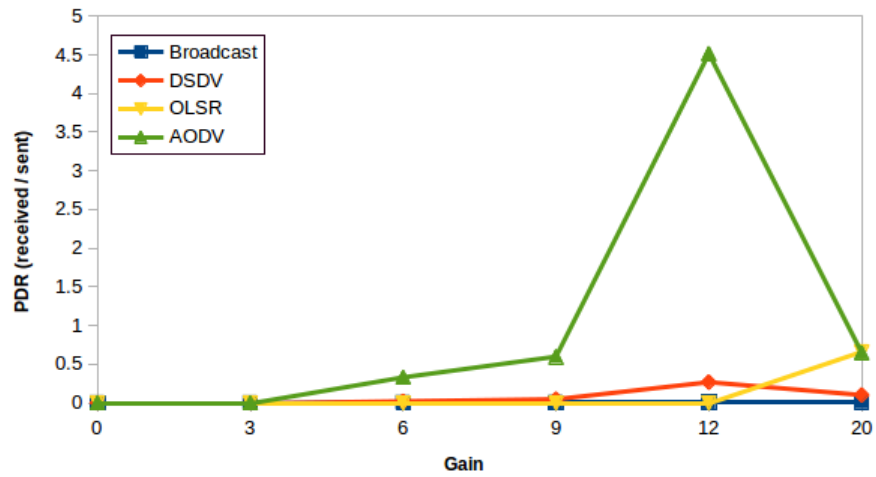Figure 4.31: PDR of All Messages by Routing Protocol

10 v 10 Swarm

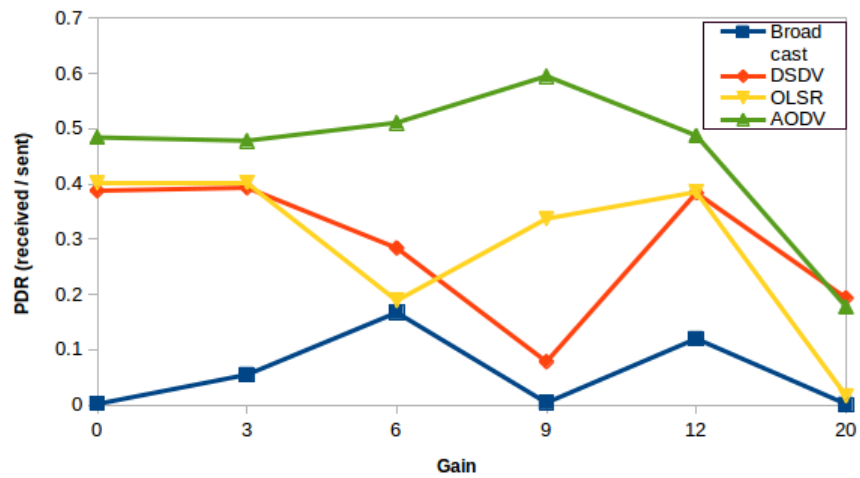Figure 4.32: PDR of All Messages by Routing Protocol

15 v 15 Swarm

Figure 4.33: PDR of All Messages by Routing Protocol

25 v 25 Swarm

CHAPTER 5:
Conclusion

5.1 Thesis Conclusion

In Chapter 4, some of the results were unexpected. No protocol consistently performed close to expected values. Those protocols that generated significant overhead created collisions or failed converge the network Not only did overhead create conditions for numerous retransmissions, but they had low message delivery metrics. The protocols with less overhead fared better.

AODV, as a reactive protocol, was able to discover multi-hop routes in restricted connectivity environments but was able to back off when nodes operated in a highly connective environment. It was able to perform across more environments than any other protocol both when delivering messages to the ground station and to nodes as evidenced by Figures 4.1 - 4.4 and 4.5 - 4.8 respectively. Even in the 15 v 15 Swarm, where many protocols failed to deliver messages to the ground, it was able to deliver some messages.

Broadcasting improved as the environment included more nodes in single-hop connections, as expected, but not in all cases. In the 10 v 10 swarm, it collapses in the 20 gain environment when sending messages to the ground, as does AODV, while DSDV excelled. It was expected that broadcasting the messages in a full single-hop environment would produce results close to expected levels but this happened infrequently. Also of note, there is a difference between ground average goodput and node average goodput. Future routing protocols may need different limits, rules, settings, or thresholds based on whether the message is traveling ground to node or node to ground.

DSDV had similar performance numbers to broadcast which implies it did not harm message delivery while updating its neighborhood routing tables but it also did not improve delivery in environments when single-hop routing was not available. OLSR, the other proactive routing protocol, was unable to deliver messages to the vehicles in all swarm sizes. The highly dynamic node topology would caused OLSR to produce numerous routing table update messages, as observed in the simulation result trace files. The DTN routing protocol,

Epidemic, produced a large number of duplicate messages in the network, also observed in the simulation trace files. Complete data was only available in the 6 v 6 swarm size. At this scale, Epidemic performed similarly to DSDV, but partial data observed in the 10 v 10 and 15 v 15 swarm sizes saw no messages reach nodes and only a limited number reach the ground. For clarity in presenting the data at swarms larger than 6 v 6, the incomplete Epidemic data was dropped but it is safe to assume poor message delivery rates will persist as the node number increases. At this time, the broadcasting method used by the NPS swarm group is working at the 50-vehicles size. It doesn't currently partition. As NPS swarm experimentation breaches its benchmarks and a routing protocol becomes necessary, it is recommended to chose a routing protocol that is either reactive in nature or is focused primarily on reducing overhead. The limited resources are quickly exhausted and overwhelm the data link layer's capacity to retransmit messages that have experienced a collision.

5.2 Future Work

This simulator takes parts from many libraries and example codes in order to model a unique swarm. The fact that the broadcast protocol does not more closely align with expected values at higher node numbers could indicate that there are still some improvements to be made even though results at low node numbers (two to five nodes) was close to expected values. Several gain levels were tested on the six v six swarm mobility file because it was the smallest swarm available at that time. The gain levels from 0 to 20 seemed to accurately represent the limited to unlimited network environments as desired. Gain settings larger than 20 produced nonsensical numbers like a single digit for the whole simulation and were thus rejected as viable settings. It may prove useful to try alternate gain ranges for each swarm size in order to accurately reproduce the environments.

The poor performance of the 15 v 15 and the subsequent improved performance from the larger 25 v 25 swarm is difficult to explain. There were also simulation runs that resulted in several nodes receiving messages while others did not over several different gain settings, as described in Chapter 4. This node split did not fall within one sub-swarm which may have flown closer to the ground station. It would be useful to compare the actual flight paths of individual nodes to their performance when only a part of a swarm receives messages and then in another environment, many more are suddenly included. The threshold could

be based on altitude rather than sub-swarm proximity to the ground station.

It would also be useful to perform the same simulation at the same swarm size but vary the mobility file. An average of performances at the same size but with different flight pathways could help factor out extreme cases based purely on lucky or unfortunate flight paths of certain vehicles. The data sets might be a more clear in-flight representation if take-off and landing flight data were excluded from the mobility models. This would produce a more clear data set of in-flight communications.

At this time, the ns-3 libraries are limited to traditional routing protocols for mobile devices. Networking researchers have had more opportunity and motivation to work with them than to network UAVs. It would be extremely beneficial to develop and add to the libraries various protocols that are more feasible or specifically designed for FANETs.

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

[1] T. H. Chung, M. R. Clement, M. A. Day, K. D. Jones, D. Davis, and M. Jones, "Live-fly, large-scale field experimentation for large numbers of fixed-wing UAVs," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1255–1262.

[2] D. T. Davis, T. H. Chung, M. R. Clement, and M. A. Day, "Multi-swarm infrastructure for swarm versus swarm experimentation," in *MILCOM 2008 - 2008 IEEE Military Communications Conference*, 2016, pp. 1–9.

[3] J. P. Rohrer, A. Jabbar, E. Perrins, and J. P. Sterbenz, "Cross-layer architectural framework for highly-mobile multihop airborne telemetry networks," in *MILCOM 2008 - 2008 IEEE Military Communications Conference*, 2008, pp. 1–9.

[4] S. Rosati, K. Kruzelecki, L. Traynard, and B. Rimoldi, "Speed-aware routing for UAV ad-hoc networks," in *2013 IEEE Globecom Workshops (GC Wkshps)*, 2013, pp. 1367–1373.

[5] I. Bekmezci, O. K. Sahingoz, and Ş. Temel, "Flying ad-hoc networks (FANETs): A survey," *Ad Hoc Networks*, vol. 11, no. 3, pp. 1254–1270, 2013.

[6] K. S. Pathapati, T. A. N. Nguyen, and J. P. Rohrer, "Performance analysis of the AeroTP transport protocol for highly-dynamic airborne telemetry networks," in *International Telemetering Conference Proceedings*, 2011.

[7] G. Hoffmann, D. G. Rajnarayan, S. L. Waslander, D. Dostal, J. S. Jang, and C. J. Tomlin, "The stanford testbed of autonomous rotorcraft for multi agent control (star-mac)," in *The 23rd Digital Avionics Systems Conference*, vol. 2. IEEE, 2004, pp. 12–E.

[8] J. Han, Y. Xu, L. Di, and Y. Chen, "Low-cost multi-UAV technologies for contour mapping of nuclear radiation field," *Journal of Intelligent & Robotic Systems*, vol. 70, no. 1, pp. 401–410, April 2013.

[9] G. Vásárhelyi, C. Virágh, G. Somorjai, N. Tarcai, T. Szörényi, T. Nepusz, and T. Vicsek, "Outdoor flocking and formation flight with autonomous aerial robots," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*. IEEE, 2014, pp. 3866–3873.

[10] J. Z. S. Hauert, S. Leven and D. Floreano. The swarming micro air vehicle network (SMAVNET) project. [Online]. Available: http://lis2.epfl.ch/CompletedResearchProjects/SwarmingMAVs/. Accessed March 27, 2017.

[11] K. Kaplan. (2016). Disney makes history with first U.S. light show powered by 300 intel drones. [Online]. Available: http://iq.intel.com/drone-light-show-takes-disney-world-visitors-future/. Accessed February 10, 2017.

[12] J. Macker, "Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations," RFC 2501, January 1999. [Online]. Available: https://www.ietf.org/rfc/rfc2501.txt

[13] K. Fall, K. L. Scott, S. C. Burleigh, L. Torgerson, A. J. Hooke, H. S. Weiss, R. C. Durst, and V. Cerf, "Delay-tolerant networking architecture," RFC 4838, April 2007. [Online]. Available: https://tools.ietf.org/html/rfc4838

[14] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *ACM SIGCOMM computer communication review*, vol. 24, no. 4. ACM, 1994, pp. 234–244.

[15] D. Johnson, Y.-c. Hu, and D. Maltz, "The dynamic source routing protocol (DSR) for mobile ad hoc networks for ipv4," RFC 4728, February 2007. [Online]. Available: https://www.ietf.org/rfc/rfc4728.txt

[16] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," RFC 3561, July 2003. [Online]. Available: https://www.ietf.org/rfc/rfc3561.txt

[17] T. Clausen and P. Jacquet, "Optimized link state routing protocol (olsr)," RFC 3626, October 2003. [Online]. Available: https://www.ietf.org/rfc/rfc3626.txt

[18] M. J. Alenazi, Y. Cheng, D. Zhang, and J. P. Sterbenz, "Epidemic routing protocol implementation in ns-3," in *Proceedings of the 2015 Workshop on ns-3*. ACM, 2015, pp. 83–90.

[19] A. Vahdat, D. Becker *et al.*, "Epidemic routing for partially connected ad hoc networks," 2000.

[20] J. Burgess, B. Gallagher, D. Jensen, B. N. Levine *et al.*, "Maxprop: Routing for vehicle-based disruption-tolerant networks." in *IEEE Infocom*, April 2006.

[21] S. D. Han and Y. W. Chung, "An improved prophet routing protocol in delay tolerant network," *The Scientific World Journal*, vol. 2015, 2015.

[22] J. P. Rohrer and K. M. Killeen, "Geolocation assisted routing protocols for vehicular networks," in *2016 International Conference on Connected Vehicles and Expo (ICCVE)*. IEEE, 2016, pp. 1–6.

[23] N. Beijar, "Zone routing protocol (ZRP)," *Networking Laboratory, Helsinki University of Technology, Finland*, vol. 9, pp. 1–12, 2002.

[24] The ns-3 simulator. [Online]. Available: http//:nsnam.org. Accessed March 21, 2017.

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California